PROGRAM AND PROCEEDINGS

IberSPEECH2018
BARCELONA
NOVEMBER 21-23
Contents

Welcome Message iii

Committees vi
  Organizing Committee vi
  Program Committee viii

Organizing Institutions xi

Awards xii

Venue xiii

Social Program xvi

Invited Speakers xix
  Tanja Schultz xix
  Rob Clark xxi
  Lluís Màrquez xxiii

Technical Program xxv

Papers xxxiii

Author’s Index 307
Welcome to IberSPEECH2018 in Barcelona, November 21-23rd 2018, co-organized by Telefónica Research, the Center for Language and Speech Technologies and Applications (TALP) at the Universitat Politècnica de Catalunya (UPC), and the Research Group on Media Technologies (GTM) at La Salle, Universitat Ramon Llull. The current edition has received inestimable valued support by the Spanish Thematic Network on Speech Technology (RTTH), the Catedra RTVE and the Voice Input Voice Output Laboratory (Vivolab) at Universidad de Zaragoza, and the ISCA Special Interest Group on Iberian Languages (SIG-IL). In addition, and for the very first time, IberSPEECH becomes an official ISCA Supported Event.

The IberSPEECH2018 event — the fourth of its kind using this name — brings together the 10th Jornadas en Tecnologías del Habla and the 6th Iberian SLTech Workshop events, aiming to promote interaction and discussion among junior and senior researchers in the field of speech and language processing for Iberian languages.

Barcelona is a modern capital of 1.7 million people and the sixth-most populous urban area in Europe. It is the home of many points of interest declared as World Heritage Sites by UNESCO such as Sagrada Familia, Park Güell, and Palau de la Musica Catalana, and also the birthplace for great minds like Antoni Gaudí, Joan Miró, Montserrat Caballé, Eduardo Mendoza, and many more. Barcelona offers a unique combination of landscapes and weather, coupled with exquisite gastronomical experiences that are the result of a blend of heritage, produce, terroir, tradition, creativity, and innovation.

The venue, the Telefónica Diagonal ZeroZero Tower by Spanish architects EMBA, is located at the very beginning of the famous Diagonal Avenue, which crosses Barcelona diagonally from the sea to the Llobregat river. The position of the Diagonal ZeroZero Tower is exceptional; it is very visible from the city and the coast, and it lays on the border between the consolidated city and the large expanses of public space in the Forum area. In fact, the address of the tower is Diagonal Avenue, number 0 and is just next to the Forum Building. The Forum area is next to the sea and the Besòs River end. It is a new business centre, located in a reformed area. Torre Telefónica is hosting several Telefónica Group companies as well as the Telefónica Research group, a leading industrial research lab following an open research model in collaboration with Universities and other research institutions. It promotes the dissemination of scientific results both through publications in top-tier peer-reviewed international journals, conferences, and technology transfer.

Following with the tradition of previous editions, IberSPEECH 2018 will be a three-day event, bringing together the best researchers and practitioners in speech and language technologies in Iberian languages to promote interaction and discussion. The organizing committee has planned a wide variety of scientific and social activities, including technical paper presen-
The core of the scientific program of IberSPEECH2018 includes a total of 37 full regular paper contributions that will be presented distributed among 5 oral and 1 poster sessions. To ensure the quality of all the contributions, each submitted paper was reviewed by three members of the scientific review committee. All the papers in the conference will be accessible through the International Speech Communication Association (ISCA) Online Archive. Paper selection was based on the scores and comments provided by the scientific review committee, which includes over 86 researchers from different institutions (mainly from Spain and Portugal, but also from France, Germany, Brazil, Slovakia, Ireland, Greece, Hungary, Slovenia, Austria and United Kingdom).

Furthermore, it is confirmed to publish an extension of selected papers as a special issue of the Journal of Applied Sciences, “IberSPEECH 2018: Speech and Language Technologies for Iberian Languages”, published by MDPI with fully open access. In addition to regular paper sessions, the IberSPEECH2018 scientific program features the following activities: the ALBAYZIN evaluation challenge session, a special session including the presentation of demos, research projects and recent PhD thesis, a round table and three keynote lectures.

Following the success of previous ALBAYZIN technology evaluations since 2006, this year ALBAYZIN evaluations have focused around multimedia analysis of TV broadcast content. Under the framework of a newly created Cátedra RTVE at Universidad de Zaragoza, we introduce and report on the results of the IberSPEECH-RTVE 2018 Challenge. The Corporación de Radiotelevisión Española (RTVE) has provided participants with an annotated TV broadcast database and the necessary tools for the evaluations, promoting the fair and transparent comparison of technology in different fields related to speech and language technology. It comprises four different challenge evaluations: Speech to Text Challenge (S2TC), Speaker Diarization Challenge (SDC) and Multimodal Diarization Challenge (MDC), organized by RTVE and Universidad de Zaragoza; and the Search on Speech Challenge (SoSC) jointly organized by Universidad San Pablo-CEU and AuDiAS from Universidad Autónoma de Madrid with the support of the ALBAYZIN Committee. Overall, 7 teams participated in the S2TC challenge, 8 teams in the SDC, 3 teams in the MDC, and 3 more teams in the SoSC challenge, which results in 21 system paper description contributions. Additionally, 11 special session papers are also included in the conference program. These were intended to describe either progress in current or recent research and development projects, demonstration systems, or PhD Thesis extended abstracts to compete in the PhD Award. Furthermore, IberSPEECH2018 features 3 remarkable keynote speakers: Prof. Tanja Schultz (University of Bremen, Germany and Institute of Carnegie Mellon, Pittsburgh, PA USA), Dr. Rob Clark (Google, London, UK) and Dr. Lluis Marquez (Amazon, Barcelona, Spain), to whom we would like to acknowledge for their extremely valuable participation.

Moreover, a round table with recognized experts brought discussion about the role of research and innovation from both academia and industry. Such a symbiosis creates market power by exploring and developing new categories which will eventually become the next blue oceans of our society. However, converting such activities into real businesses, making strong bones and figuring out new products that have a major impact in the real world is a non trivial task. A round table, we expected as an opportunity for both worlds on finding and exploring synergies and collaboration.
The social program of IberSPEECH2018 sets sail with the welcome reception at the Escola d’Enginyeria Barcelona Est (EEBE), at the recently created UPC Diagonal Besòs Campus, next to the Telefonica Tower. EEBE aims to become a top-quality academic centre in the field of engineering for the 21st-century industry that is capable of acting as an agent of transformation at a local and international level. The EEBE was born from the Barcelona College of Industrial Engineering (EUETIB) and from part of the teaching and research activity in chemical and materials engineering hitherto carried out at the Barcelona School of Industrial Engineering (ETSEIB). The gala dinner will be held at Restaurant Marítim, next to the legendary Barcelona Reial Club Marítim, designed by Lázaro Rosa-Violán from Contemporain Studio, to provide the aesthetics and flavours of different Mediterranean paradises.

Finally, we would like to thank all those whose effort made possible this conference, including the members of the organizing committee, the local organizing committee, the ALBAYZIN committee, the scientific reviewer committee, the authors, the conference attendees, the supporting institutions, and so many people who gave their best to achieve a successful conference.

Barcelona, November 2018
Jordi Luque, General Chair
Organizing Committee

**General Chair:**
Jordi Luque, Telefónica Research, Spain

**General Co-Chairs:**
Antonio Bonafonte, Universitat Politècnica de Catalunya, Spain
Francesc Alíás Pujol, La Salle – Universitat Ramon LLull, Spain
António Teixeira, Universidade de Aveiro, Portugal

**Technical Program Chair:**
Javier Hernando Pericás, Universitat Politècnica de Catalunya, Spain

**Technical Program Co-Chairs:**
Alberto Abad, INESC ID Lisboa / IST Lisboa, Portugal
Xavier Anguera, ELSA Corp., United States
Carlos David Martínez Hinarejos, Universitat Politècnica de València, Spain
Eva Navas, University of the Basque Country, UPV- EHU, Spain
Carlos Segura, Telefónica Research, Spain

**Publication Chairs:**
Francesc Alíás Pujol, La Salle – Universitat Ramon LLull, Spain
Antonio Bonafonte, Universitat Politècnica de Catalunya, Spain
**Special Session and Awards Chair:**
Ascensión Gallardo-Antolín, Universidad Carlos III, Spain

**Plenary Talks & Round Table Chairs:**
Joan Serrà, Telefónica Research, Spain  
Marta Ruiz Costa-Jussà, Universitat Politècnica de Catalunya, Spain

**Evaluations Chairs:**
Alfonso Ortega, Universidad de Zaragoza, Spain  
Eduardo Lleida, Universidad de Zaragoza, Spain  
Luis Javier Rodríguez Fuentes, Universidad del País Vasco, Spain

**Local Committee:**
Francesc Alías Pujol, La Salle – Universitat Ramon LLull, Spain  
Antonio Bonafonte, Universitat Politècnica de Catalunya, Spain  
Jordi Luque, Telefónica Research, Spain  
Jordi Pons, Universitat Pompeu Fabra, Spain  
Bardia Rafieian, Universitat Politècnica de Catalunya, Spain  
Marta Ruiz Costa-Jussà, Universitat Politècnica de Catalunya, Spain  
Carlos Segura, Telefónica Research, Spain  
Joan Serrà, Telefónica Research, Spain
Scientific Review Committee

Alberto Abad, INESC-IST, Portugal
Francesc Àlias, La Salle – Universitat Ramon Llull, Spain
Aitor Alvarez, Vicomtech-IK4, Spain
Xavier Anguera, Miro ELSA Corp., Portugal
Jorge Baptista, INESC-ID Lisboa, Portugal
Plínio Barbosa, University of Campinas, Brazil
Fernando Batista, INESC-ID & ISCTE-IUL, Portugal
José Miguel Benedí, Universitat Politècnica de València, Spain
Antonio Bonafonte, Universitat Politècnica de Catalunya, Spain
João Cabral, Trinity College Dublin, Ireland
Francisco Casacuberta, Universitat Politècnica de València, Spain
María José Castro-Bleda, Universitat Politècnica de València, Spain
Ricardo Córdoba, Universidad Politécnica de Madrid, Spain
Conceição Cunha, IPS Munich, Germany
Carme de-la-Mota, Universitat Autònoma de Barcelona, Spain
Arantza del Pozo, Vicomtech, Spain
Laura Docío-Fernández, University of Vigo, Spain
Daniel Erro, Cirrus Logic, Madrid, Spain
David Escudero, University of Valladolid, Spain
Nicholas Evans, EURECOM, France
Mireia Farrús, Universitat Pompeu Fabra, Spain
Rubén Fernández, Universidad Politécnica de Madrid, Spain
Javier Ferreiros, Universidad Politécnica de Madrid, Spain
Julian Fierrez, Universidad Autonoma de Madrid, Spain
Ascensión Gallardo, Universidad Carlos III de Madrid, Spain
Fernando García Granada, Universitat Politècnica de València, Spain
Carmen García Mateo, University of Vigo, Spain
Kafentzis George, University of Crete, Greece
Omid Ghahabi, EML European Media Laboratory GmbH, Germany
Juna Ignacio Godino, Llorente Universidad Politécnica de Madrid, Spain
Jon Sánchez, Aholab – EHU/UPV, Spain
Joan Andreu, Sanchez Universitat Politècnica de València, Spain
Emilio Sanchis, Universitat Politècnica de València, Spain
Diana Santos, University of Oslo, Norway
Ibon Saratxaga, University of the Basque Country, Spain
Encarna Segarra, Universitat Politècnica de València, Spain
Carlos Segura Perales, Telefónica Research, Spain
Joan Serrà, Telefónica Research, Spain
Alberto Simões, 2Ai Lab – IPCA, Portugal
Rubén Solera-Ureña, INESC-ID Lisboa, Portugal
António Teixeira, University of Aveiro, Portugal
Javier Tejedor, Universidad CEU San Pablo, Spain
Doroteo Toledano, Universidad Autónoma de Madrid, Spain
Isabel Trancoso, INESC ID Lisboa / IST, Portugal
Cassia Valentini-Botinhao, The University of Edinburgh, UK
Amparo Varona, University of the Basque Country, Spain
Andrej Zgank, University of Maribor, Slovenia
Catalin Zorila, Toshiba Cambridge Research Laboratory UK
Organizing Institutions

This conference has been organized by:

Telefónica  
UNIVERSITAT POLITÈCNICA DE CATALUNYA  
laSalle

with the collaboration of:

RTTH  
Sca  
Sgl  
Universidad Zaragoza  
RTVE

IberSPEECH 2018 has been partially funded by the project Red Temática en Tecnologías del Habla 2017 (TEC2017-90829-REDT) founded by Ministerio de Ciencia, Innovación y Universidades.
Awards

Best Paper Award

All regular papers are candidates for this award. The award, given based on the review reports and the presentation at the conference, grants the authors the publication of an extended version of their work within the Special Issue of Applied Sciences journal (MDPI) entitled “IberSPEECH 2018: Speech and Language Technologies for Iberian Languages”.

Best Albayzin evaluation system

Papers submitted to Albayzin evaluation tasks are candidates for these awards. The awards will be given to the winners of the Albayzin evaluation challenges, in accordance with the evaluation plan and rules defined for each task.

Best PhD Thesis award

Papers submitted to the PhD Thesis special session are candidates for this award. The award is given based on the decision of the committee formed by the members of the General chair, Technical Program chair and Special Session and Awards Chair. The award is given based on different criteria, including the quality of the document, impact of the thesis and clearness of the presentation at the conference.

Professional Career Prize in Speech Technologies

This is an honorary prize awarded by the Spanish Thematic Network on Speech Technology (RTTH) that recognizes experienced individuals who have made outstanding contributions related to speech technology research in Spain.

IberSPEECH2018 Edition

- José B. Mariño Acebal, Universitat Politècnica de Catalunya
- Antonio Rubio Ayuso, Universidad de Granada.
Main conference venue

Torre Telefónica

Address:
Torre Telefónica – Diagonal 00
Plaza de Ernest Lluch i Martín, 5
08019 Barcelona – Spain

WI-FI access instructions

Connect to IberSPEECH 2018’s network with the following password:

- Network: Iberspeech 2018
- Password: *******
The main body of the conference will be held in Torre Telefónica (floor 0, see figure) and the Auditorium in floor 2 by accessing the elevators depicted in the figure. The following diagram outlines the main conference areas and services:
D’Ins Escola, Restaurant i Càtering

A gastronomic offer adapted to the occasion and a catering service with an added value that will enrich it: THE SOCIAL VALUE of the PEOPLE that work in this service. People who participate in a training and job placement program developed by the Fundación Formació i Treball.

Address: Carrer de Ramon Llull, 240, 08930, Sant Adrià del Besòs.

Wednesday 21st and Thursday 22nd lunches’ will be given very close to Torre Telefónica (350 m).
Welcome Reception: Wednesday 21 November, 20:00.

Escola d’Enginyeria Barcelona Est (UPC Diagonal Besòs)
Address: Av. Eduard Maristany, 16, 08019, Barcelona

The welcome reception will be given at the Rambla de Colors space in UPC Diagonal Besòs (230 m from Torre Telefónica). The space is placed between buildings C and I being possible to access it from both main entrances at each building. Use the number 37400 at the entrance phone to connect with reception and granting access to the building and go down to the underground level.

Restaurant Maritim
Address: Moll d’Espanya, 08039, Barcelona
Telephone: +34 93 221 17 75
Web: www.maritimrestaurant.es

The gala dinner will be held downtown, next to the sea (close to Cristobal Colon statue).

How do I get there?
Take L4 from El Maresme – Fòrum metro station to Barceloneta.
You can also reach the restaurant by walking the seaside (1h 15 min).
Tanja Schultz received her diploma and doctoral degree in Informatics from University of Karlsruhe, Germany, in 1995 and 2000. Prior to these degrees she completed the state exam in Mathematics, Sports, Physical and Educational Science from Heidelberg University, Germany in 1989. She is currently the Professor for Cognitive Systems at the University of Bremen, Germany and adjunct Research Professor at the Language Technologies Institute of Carnegie Mellon, PA USA. Since 2007, she directs the Cognitive Systems Lab, where her research activities include multilingual speech recognition and the processing, recognition, and interpretation of biosignals for human-centered technologies and applications. Prior to joining University of Bremen, she was a Research Scientist at Carnegie Mellon (2000-2007) and a Full Professor at Karlsruhe Institute of Technology in Germany (2007-2015). Dr. Schultz is an Associate Editor of ACM Transactions on Asian Language Information Processing (since 2010), serves on the Editorial Board of Speech Communication (since 2004), and was Associate Editor of IEEE Transactions on Speech and Audio Processing (2002-2004). She was President (2014-2015) and elected Board Member (2006-2013) of ISCA, and a General Co-Chair of Interspeech 2006. She was elevated to Fellow of ISCA (2016) and to member of the European Academy of Sciences and Arts (2017). Dr. Schultz was the recipient of the Otto Haxel Award in 2013, the Alcatel Lucent Award for Technical Communication in 2012, the PLUX Wireless Biosignals Award in 2011, and the Allen Newell Medal for Research Excellence in 2002, and received the ISCA / EURASIP Speech Communication Best paper awards in 2001 and 2015.

Abstract.- Speech is a complex process emitting a wide range of biosignals, including, but not limited to, acoustics. These biosignals – stemming from the articulators, the articulator muscle activities, the neural pathways, and the brain itself – can be used to circumvent limitations of conventional speech processing in particular, and to gain insights into the process of speech production in general. In my talk I will present ongoing research at the Cognitive Systems Lab (CSL), where we explore a variety of speech-related muscle and brain activi-
ties based on machine learning methods with the goal of creating biosignal-based speech processing devices for communication applications in everyday situations and for speech rehabilitation, as well as gaining a deeper understanding of spoken communication. Several applications will be described such as Silent Speech Interfaces that rely on articulatory muscle movement captured by electromyography to recognize and synthesize silently produced speech, Brain-to-text interfaces that recognize continuously spoken speech from brain activity captured by electrocorticography to transform it into text, and Brain-to-Speech interfaces that directly synthesize audible speech from brain signals.
Rob Clark received his PhD from the University of Edinburgh in 2003. His primary interest is in producing engaging synthetic speech. Before joining Google Rob was at the University of Edinburgh for many years involved in both teaching and research relating to text-to-speech synthesis. Rob was one of the primary developers and maintainers of the open source Festival text-to-speech synthesis system. In 2015 he joined Google where he is working on text-to-speech synthesis and prosody.

Abstract.- This talk addresses the issue of producing appropriate and engaging text-to-speech. The quality of speech produced by modern text-to-speech systems is sufficiently intelligible and naturally sounding that we are now seeing it widely used in an increasing number of real world applications. While the speech generated can sound very natural, we are still a long way from ensuring it always sounds appropriate and engaging in the context of a particular discourse or dialogue. We present recent work at Google which begins to address this issue by looking at techniques to generate variation in prosody and speaking style using latent representations and discuss the problems and challenges that we face in going further.
Abstract.- Automatic Question Answering (Q&A), i.e., the task of building computer programs that are able to answer question posed in natural language, has a long tradition in the fields of Natural Language Processing and Information Retrieval. In recent years, Q&A applications have had a tremendous impact in industry and they are ubiquitous (e.g., embedded in any of the personal assistants that are in the market, Siri, Alexa, Cortana, Google Assistant, etc.). At the same time, we have witnessed a renewed interest in the scientific community, as Q&A has become one of the paradigmatic tasks for assessing the ability of machines to comprehend text. A plethora of corpora, resources and systems have blossomed and flooded the community in the last three years. These systems can do very impressive things, for instance, finding answers to open ended questions in long text contexts with super-human accuracy, or answering complex questions about images, by mixing the two modalities. As in many other fields, these state-of-the-art systems are implemented using machine learning in the form of neural networks (deep learning). The new AI, of course. But do these Q&A systems really understand what they read? In more simple words, do they provide the right answers for the right reasons? Several recent studies have shown that QA systems are actually very brittle. They generalize badly and they fail miserably when presented with simple adversarial examples. The machine learning algorithms are very good at picking all the biases and
artefacts in the corpora, and they learn to find answers based on shallow text properties and pattern matching. But they do not show many understanding or reasoning abilities, after all. Following this serious setback, there is a new push in the community for carefully designing more complex and bias-free datasets, and more robust and explainable systems. Hopefully, this will lead to a new generation of smarter and more useful Q&A engines in the near future. In this talk, I will overview the present and the future of Question Answering by going over all the aforementioned topics.
Technical Program

**Speaker Recognition**

**Wednesday, 21 November 2018, 09:20 – 10:40**

**Chair:** Xavier Anguera, ELSA Corp.

Differentiable Supervector Extraction for Encoding Speaker and Phrase Information in Text Dependent Speaker Verification  
Victoria Mingote, Antonio Miguel, Alfonso Ortega, Eduardo Lleida  

Phonetic Variability Influence on Short Utterances in Speaker Verification  
Ignacio Viñals, Alfonso Ortega, Antonio Miguel, Eduardo Lleida  

Restricted Boltzmann Machine Vectors for Speaker Clustering  
Umair Khan, Pooyan Safari, Javier Hernando  

Speaker Recognition under Stress Conditions  
Esther Rituerto-González, Ascensión Gallardo-Antolín, Carmen Peláez-Moreno

**Topics on Speech Technologies**

**Wednesday, 21 November 2018**  
**12:00 – 13:30**

**Chair:** Alberto Abad, INESC-ID/IST

Bilingual Prosodic Dataset Compilation for Spoken Language Translation  
Alp Öktem, Mireia Farrús, Antonio Bonafonte  

Building an Open Source Automatic Speech Recognition System for Catalan  
Baybars Külebi, Alp Öktem  

Multi-Speaker Neural Vocoder  
Oriol Barbany, Antonio Bonafonte, Santiago Pascual  

Improving the Automatic Speech Recognition through the improvement of Language Models  
Andrés Piñeiro-Martín, Carmen García-Mateo, Laura Docío-Fernández
Towards expressive prosody generation in TTS for reading aloud applications
Monica Dominguez, Alicia Burga, Mireia Farrús, Leo Wanner

Performance evaluation of front- and back-end techniques for ASV spoofing detection systems based on deep features
Alejandro Gomez-Alanis, Antonio M. Peinado, José Andrés González López, Angel M. Gomez

The observation likelihood of silence: analysis and prospects for VAD applications
Igor Odriozola, Inma Hernaez, Eva Navas, Luis Serrano, Jon Sanchez

On the use of Phone-based Embeddings for Language Recognition
Christian Salamea, Ricardo de Córdoba, Luis Fernando D’Haro, Rubén San-Segundo, Javier Ferreiros

End-to-End Speech Translation with the Transformer
Laura Cross Vila, Carlos Escolano, José A. R. Fonollosa, Marta R. Costa-Jussà

Audio event detection on Google’s Audio Set database: Preliminary results using different types of DNNs
Javier Darna-Sequeiros, Doroteo T. Toledano

Emotion Detection from Speech and Text
Mikel de Velasco, Raquel Justo, Josu Antón, Mikel Carrilero, M. Inés Torres

Experimental Framework Design for Sign Language Automatic Recognition
Darío Tilves Santiago, Ian Benderitter, Carmen García-Mateo

Baseline Acoustic Models for Brazilian Portuguese Using Kaldi Tools
Cassio Batista, Ana Larissa Dias, Nelson Sampaio Neto
### ASR & Speech Applications

**Wednesday, 21 November 2018, 15:00 – 16:40**

**Chair: Carmen García Mateo, University of Vigo**

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converted Mel-Cepstral Coefficients for Gender Variability Reduction in Query-by-Example Spoken Document Retrieval</td>
<td>82</td>
</tr>
<tr>
<td>Paula López Otero, Laura Docío-Fernández</td>
<td></td>
</tr>
<tr>
<td>A Recurrent Neural Network Approach to Audio Segmentation for Broadcast Domain Data</td>
<td>87</td>
</tr>
<tr>
<td>Pablo Gimeno, Ignacio Viñals, Alfonso Ortega, Antonio Miguel, Eduardo Lleida</td>
<td></td>
</tr>
<tr>
<td>Improving Transcription of Manuscripts with Multimodality and Interaction</td>
<td>92</td>
</tr>
<tr>
<td>Emilio Granell, Carlos David Martinez Hinarejos, Verónica Romero</td>
<td></td>
</tr>
<tr>
<td>Improving Pronunciation of Spanish as a Foreign Language for L1 Japanese Speakers with Japañol CAPT Tool</td>
<td>97</td>
</tr>
<tr>
<td>Cristian Tejedor-García, Valentín Cardeñoso-Payo, María J. Machuca, David Escudero-Mancebo, Antonio Ríos, Takuya Kimura</td>
<td></td>
</tr>
<tr>
<td>Exploring E2E speech recognition systems for new languages</td>
<td>102</td>
</tr>
<tr>
<td>Conrad Bernath, Aitor Alvarez, Haritz Arzelus, Carlos David Martinez</td>
<td></td>
</tr>
</tbody>
</table>
Speech & Language Technologies Applied to Health
Wednesday, 21 November 2018, 17:00 – 18:40
Chair: Mireia Farrús, Universitat Pompeu Fabra

Listening to Laryngectomees: A study of Intelligibility and Self-reported Listening Effort of Spanish Oesophageal Speech
Sneha Raman, Inma Hernaez, Eva Navas, Luis Serrano

Towards an automatic evaluation of the prosody of people with Down syndrome
Mario Corrales-Astorgano, Pastora Martinez-Castilla, David Escudero-Mancebo, Lourdes Aguilar, César González-Ferreras, Valentín Cardeñoso-Payo

Whispered-to-voiced Alaryngeal Speech Conversion with Generative Adversarial Networks
Santiago Pascual, Antonio Bonafonte, Joan Serrà, José Andrés González López

LSTM based voice conversion for laryngectomees
Luis Serrano, David Tavarez, Xabier Sarasola, Sneha Raman, Ibon Saratxaga, Eva Navas, Inma Hernaez

Sign Language Gesture Classification using Neural Networks
Zuzanna Parcheta, Carlos David Martinez Hinarejos

Synthesis, Production & Analysis
Thursday, 22 November 2018, 09:00 – 10:40
Chair: Francesc Alías Pujol, La Salle - Universitat Ramon Llull

Influence of tense, modal and lax phonation on the three-dimensional finite element synthesis of vowel [A]
Marc Freixes, Marc Arnela, Joan Claudi Socoró, Francesc Alías Pujol, Oriol Guasch

Exploring Advances in Real-time MRI for Speech Production Studies of European Portuguese
Conceicao Cunha, Samuel Silva, António Teixeira, Catarina Oliveira, Paula Martins, Arun Joseph, Jens Frahm

A postfiltering approach for dual-microphone smartphones
Juan M. Martín-Doñas, Iván López-Espejo, Angel M. Gomez, Antonio M. Peinado

Speech and monophonic singing segmentation using pitch parameters
Xabier Sarasola, Eva Navas, David Tavarez, Luis Serrano, Ibon Saratxaga

Self-Attention Linguistic-Acoustic Decoder
Santiago Pascual, Antonio Bonafonte, Joan Serrà
Special Session
Thursday, 22 November 2018 12:00 – 13:30
Chair: Ricardo de Córdoba, Universidad Politécnica de Madrid

Show & Tell

Japañol: a mobile application to help improving Spanish pronunciation by Japanese native speakers
Cristian Tejedor-García, Valentín Cardeñoso-Payo, David Escudero-Mancebo

Ongoing Research Projects

Towards the Application of Global Quality-of-Service Metrics in Biometric Systems
Juan Manuel Espín, Roberto Font, Juan Francisco Inglés-Romero, Cristina Vicente-Chicote

Incorporation of a Module for Automatic Prediction of Oral Productions Quality in a Learning Video Game
David Escudero-Mancebo, Valentín Cardeñoso-Payo

Silent Speech: Restoring the Power of Speech to People whose Larynx has been Removed
José Andrés González López, Phil D. Green, Damian Murphy, Amelia Gully, James M. Gilbert

RESTORE Project: REpair, STOrage and REhabilitation of speech
Inma Hernaez, Eva Navas, Jose Antonio Municio Martín, Javier Gomez Suárez

Corpus for Cyberbullying Prevention
Asuncion Moreno, Antonio Bonafonte, Igor Jauk, Laia Tarrés, Victor Pereira

EMPATHIC, Expressive, Advanced Virtual Coach to Improve Independent Healthy-Life-Years of the Elderly
M. Inés Torres, Gérard Chollet, César Montenegro, Jofre Tenorio-Laranga, Olga Gordeeveva, Anna Esposito, Cornelius Glackin, Stephan Schlögl, Olivier Deroo, Begoña Fernández-Ruanova, Riberto Santana, Maria S. Kornes, Fred Lindner, Daria Kyslitska, Miriam Reiner, Gennaro Cordasco, Mari Aksnes, Raquel Justo

PhD Thesis

Advances on the Transcription of Historical Manuscripts based on Multimodality, Interactivity and Crowdsourcing
Emilio Granell, Carlos David Martinez Hinarejos, Verónica Romero
Bottleneck and Embedding Representation of Speech for DNN-based Language and Speaker Recognition
Alicia Lozano-Diez, Joaquin Gonzalez-Rodriguez, Javier Gonzalez-Dominguez

Deep Learning for i-Vector Speaker and Language Recognition: A Ph.D. Thesis Overview
Omid Ghahabi

Unsupervised Learning for Expressive Speech Synthesis
Igor Jauk

Albayzin Evaluation
Thursday, 22 November 2018
15:00 – 16:40
Chair: Alfonso Ortega & Eduardo Lleida, Universidad de Zaragoza

Multimodal Diarization Challenge

ODESSA/PLUMCOT at Albayzin Multimodal Diarization Challenge 2018
Benjamin Maurice, Hervé Bredin, Ruiqing Yin, Jose Patino, Héctor Delgado, Claude Barras, Nicholas Evans, Camille Guinaudeau

UPC Multimodal Speaker Diarization System for the 2018 Albayzin Challenge
Miquel Angel India Massana, Itziar Sagastiberry, Ponç Palau, Elisa Sayrol, Josep Ramon Morros, Javier Hernando

The GTM-UVIGO System for Audiovisual Diarization
Eduardo Ramos-Muguerza, Laura Docio-Fernández, José Luis Alba-Castro

Speaker Diarization Challenge

The SRI International STAR-LAB System Description for IberSPEECH-RTVE 2018 Speaker Diarization Challenge
Diego Castan, Mitchell McLaren, Mahesh Kumar Nandwana

ODESSA at Albayzin Speaker Diarization Challenge 2018
Jose Patino, Héctor Delgado, Ruiqing Yin, Hervé Bredin, Claude Barras, Nicholas Evans

EML Submission to Albayzin 2018 Speaker Diarization Challenge
Omid Ghahabi, Volker Fischer

In-domain Adaptation Solutions for the RTVE 2018 Diarization Challenge
Ignacio Viñals, Pablo Gimeno, Alfonso Ortega, Antonio Miguel, Eduardo Lleida
DNN-based Embeddings for Speaker Diarization in the AuDiAS-UAM System for the Albayzin 2018 IberSPEECH-RTVE Evaluation  
Alicia Lozano-Diez, Beltran Labrador, Diego de Benito, Pablo Ramirez, Doroteo T. Toledano

CENATAV Voice-Group Systems for Albayzin 2018 Speaker Diarization Evaluation Campaign  
Edward L. Campbell, Gabriel Hernandez, José R. Calvo de Lara

The Intelligent Voice System for the IberSPEECH-RTVE 2018 Speaker Diarization Challenge  
Abbas Khosravani, Cornelius Glackin, Nazim Dugan, Gérard Chollet, Nigel Cannings

JHU Diarization System Description  
Zili Huang, L. Paola García-Perera, Jesús Villalba, Daniel Povey, Najim Dehak

Search on Speech Challenge

GTM-IRLab Systems for Albayzin 2018 Search on Speech Evaluation  
Paula López Otero, Laura Docío-Fernández

AUDIAS-CEU: A Language-independent approach for the Query-by-Example Spoken Term Detection task of the Search on Speech ALBAYZIN 2018 evaluation  
Maria Cabello, Doroteo T. Toledano, Javier Tejedor

GTTS-EHU Systems for the Albayzin 2018 Search on Speech Evaluation  
Luis J. Rodríguez-Fuentes, Mikel Peñagarikano, Amparo Varona, Germán Bordel

Cenatav Voice Group System for Albayzin 2018 Search on Speech Evaluation  
Ana R. Montalvo, Jose M. Ramirez, Alejandro Roble, Jose R. Calvo

Speech to Text Challenge

MLLP-UPV and RWTH Aachen Spanish ASR Systems for the IberSpeech-RTVE 2018 Speech-to-Text Transcription Challenge  
Javier Jorge, Adrià Martínez-Villaronga, Pavel Golik, Adrià Giménez, Joan Albert Silvestre-Cerdà, Patrick Doetsch, Vicent Andreu Ciscar, Hermann Ney, Alfons Juan, Albert Sanchis

Exploring Open-Source Deep Learning ASR for Speech-to-Text TV program transcription  
Juan M. Perero-Codosero, Javier Antón-Martín, Daniel Tapias Merino, Eduardo López-Gonzalo, Luis A. Hernández-Gómez
The Vicomtech-PRHLT Speech Transcription Systems for the IberSPEECH-RTVE 2018 Speech to Text Transcription Challenge
Haritz Arzelus, Aitor Alvarez, Conrad Bernath, Eneritz García, Emilio Granell, Carlos David Martinez Hinarejos

Intelligent Voice ASR system for Iberspeech 2018 Speech to Text Transcription Challenge
Nazim Dugan, Cornelius Glackin, Gérard Chollet, Nigel Cannings

The GTM-UVIGO System for Albayzin 2018 Speech-to-Text Evaluation
Laura Docío-Fernández, Carmen García-Mateo

Text & NLP Applications
Friday, 23 November 2018, 09:00 – 10:40
Chair: José F. Quesada, Universidad de Sevilla

Topic coherence analysis for the classification of Alzheimer’s disease
Anna Pompili, Alberto Abad, David Martins de Matos, Isabel Pavão Martins

Building a global dictionary for semantic technologies
Iklódi Eszter, Gábor Recski, Gábor Borbély, Maria Jose Castro-Bleda

TransDic, a public domain tool for the generation of phonetic dictionaries in standard and dialectal Spanish and Catalan
Juan-María Garrido, Marta Codina, Kimber Fodge

Wide Residual Networks 1D for Automatic Text Punctuation
Jorge Llombart, Antonio Miguel, Alfonso Ortega, Eduardo Lleida

End-to-End Multi-Level Dialog Act Recognition
Eugénio Ribeiro, Ricardo Ribeiro, David Martins de Matos
Differentiable Supervector Extraction for Encoding Speaker and Phrase Information in Text Dependent Speaker Verification

Victoria Mingote, Antonio Miguel, Alfonso Ortega, Eduardo Lleida
ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain
{vmingote,amiguel,ortega,lleida}@unizar.es

Abstract
In this paper, we propose a new differentiable neural network alignment mechanism for text-dependent speaker verification which uses alignment models to produce a supervector representation of an utterance. Unlike previous works with similar approaches, we do not extract the embedding of an utterance from the mean reduction of the temporal dimension. Our system replaces the mean by a phrase alignment model to keep the temporal structure of each phrase which is relevant in this application since the phonetic information is part of the identity in the verification task. Moreover, we can apply a convolutional neural network as front-end, and thanks to the alignment process being differentiable, we can train the whole network to produce a supervector for each utterance which will be discriminative with respect to the speaker and the phrase simultaneously. As we show, this choice has the advantage that the supervector encodes the phrase and speaker information providing good performance in text-dependent speaker verification tasks. In this work, the process of verification is performed using a basic similarity metric, due to simplicity, compared to other more elaborate models that are commonly used. The new model using alignment to produce supervectors was tested on the RSR2015-Part I database for text-dependent speaker verification, providing competitive results compared to similar size networks using the mean to extract embeddings.

Index Terms: Text Dependent Speaker verification, HMM Alignment, Deep Neural Networks, Supervectors

1. Introduction
Recently, techniques based on discriminative deep neural networks (DNN) have achieved a substantial success in many speaker verification tasks. These techniques follow the philosophy of the state-of-the-art face verification systems [1][2] where embeddings are usually extracted by reduction mechanisms and the decision process is based on a similarity metric [3]. Unfortunately, in text-dependent tasks this approach does not work efficiently since the pronounced phrase is part of the identity information [4][5]. A possible cause of the imprecision in text-dependent tasks could be derived from using the mean as a representation of the utterance as we show in the experimental section. To solve this problem, this paper shows a new architecture which combines a deep neural network with a phrase alignment method used as a new internal layer to maintain the temporal structure of the utterance. As we will show, it is a more natural solution for the text-dependent speaker verification, since the speaker and phrase information can be encoded in the supervector thanks to the neural network and the specific states of the supervector.

In the context of text-independent speaker verification tasks, the baseline system based on i-vector extraction and Probabilistic Linear Discriminant Analysis (PLDA) [6][7] are still among the best results of the state-of-the-art. The i-vector extractor represents each utterance in a low-dimensional subspace called the total variability subspace as a fixed-length feature vector and the PLDA model produces the verification scores. However, as we previously mentioned, many improvements on this baseline system have been achieved in recent years by progressively substituting components of the systems by DNNs, thanks to their larger expressiveness and the availability of bigger databases. Examples of this are the use of DNN bottleneck representations as features replacing or combined with spectral parametrization [8], training DNN acoustic models to use their outputs as posteriors for alignment instead of GMMs in i-vector extractors [9], or replacing PLDA by a DNN [10]. Other proposals similar to face verification architectures have been more ambitious and have trained a discriminative DNN for multiclass classifying and then extract embeddings by reduction mechanisms [11][12], for example taking the mean of an intermediate layer named usually bottleneck layer. After that embedding extraction, the verification score is obtained by a similarity metric such as cosine similarity [11].

The application of DNNs and the same techniques as in text-independent models for text-dependent speaker verification tasks has produced mixed results. On the one hand, specific modifications of the traditional techniques have been shown successful for text-dependent tasks such as i-vectors+PLDA [13], DNNs bottleneck as features for i-vector extractors [14] or posterior probabilities for i-vector extractors [14][15]. On the other hand, speaker embeddings obtained directly from a DNN have provided good results in tasks with large amounts of data and a single phrase [16] but they have not been as effective in tasks with more than one pass phrase and smaller database sizes [4][5]. The lack of data in this last scenario may lead to problems with deep architectures due to overfitting of models.

Another reason that we explore in the paper for the lack of effectiveness of these techniques in general text-dependent tasks is that the phonetic content of the uttered phrase is relevant for the identification. State-of-art text-independent approaches to obtain speaker embeddings from an utterance usually reduce temporal information by pooling and by calculating the mean across frames of the internal representations of the network. This approach may neglect the order of the phonetic information because in the same phrase the beginning of the sentence may be totally different from what is said at the end. An example of this is the case when the system asks the speaker to utter digits in some random order. In that case a mean vector would fail to capture the combination of phrase and speaker. Therefore one of the objectives of the paper is to show that it is important to keep this phrase information for the identification process, not just the information of who is speaking.

In previous works we have developed systems that need to store a model per user which were adapted from a universal background model and the evaluation of the trial was based on
a likelihood ratio \[17\]. One of the drawbacks of this approach is the need to store a large amount of data per user and the speed of evaluation of trials, since likelihood expressions were dependent on the frame length. In this paper, we focus on systems using a vector representation of a trial or a speaker model. We propose a new approach that includes alignment as a key component of the mechanism to obtain the vector representation from a deep neural network. Unlike previous works, we substitute the mean of the internal representations across time which is used in other neural network architectures \[4\]\[5\] by a frame to state alignment to keep the temporal structure of each utterance. We show how the alignment can be applied in combination with a DNN acting as a front-end to create a supervector for each utterance. As we will show, the application of both sources of information in the process of defining the supervector provides better results in the experiments performed on RSR2015 compared to previous approaches.

This paper is organized as follows. In Section 2 we present our system and especially the alignment strategy developed. Section 3 presents the experimental data. Section 4 explains the results achieved. Conclusions are presented in Section 5.

2. Deep neural network based on alignment

In view of the aforementioned imprecisions in the results achieved in previous works for this task with only DNNs and a basic similarity metric, we decided to apply an alignment mechanism due to the importance of the phrases and their temporal structure in this kind of tasks. Since same person does not always pronounce one phrase at the same speed or in the same way due to differences in the phonetic information, it is usual that there exists an articulation and pronunciation mismatch between two compared speech utterances even from the same person.

In Fig. 1 we show the overall architecture of our system, where the mean reduction to obtain the vector embedding before the backend is substituted by the alignment process to finally create a supervector by audio file. This supervector can be seen as a mapping between an utterance and the state components of the alignment, which allows to encode the phrase information. For the verification process, once our system is trained, one supervector is extracted for each enroll and test file, and then a cosine metric is applied over them to achieve the verification scores.

Figura 1: Differentiable neural network alignment mechanism based on alignment models. The supervector is composed of \(Q\) vectors \(s_q\) for each state.

\[\gamma = [1, 1, 1, 2, 2, 3, 3, 4] \rightarrow A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)\]

After this process, as we show in Fig. 2, we added this matrix to the network as a matrix multiplication like one layer before, thanks to the expression as a matrix product it is easy to differentiate and this enables to backpropagate gradients to train neural network as usual. This matrix multiplication allows assigning the corresponding frames to each state resulting in a supervector. Then, the speaker verification is performed with this supervector. The alignment as a matrix multiplication can be expressed as a function of the input signal to this layer \(x_{ct}\) with dimensions \((c \times t)\) and matrix of alignment of each utterance \(A\) with dimensions \((t \times q)\):

\[s_{cq} = \frac{\sum_t x_{ct} \cdot a_{c}}{\sum_t a_{c}} \quad (2)\]

where \(s_{cq}\) is the supervectors with dimensions \((c \times q)\), where there are \(q\) state vectors of dimension \(c\) and we normalize with the number of times state \(q\) is activated.

2.2. Deep neural network architecture

As a first approximation to check that the previous alignment layer works better than extracting the embedding from the mean reduction, we apply this mentioned layer directly on the
Figura 2: Process of alignment, the input signal $x$ is multiplied by an alignment matrix $A$ to produce a matrix with vectors $s_Q$ which are then concatenated to obtain the supervector.

input signal over the acoustic features thus we obtain the traditional supervisor. However, we expect to improve this baseline result, so we propose to add some layers as front-end previous to the alignment layer and train them in combination with the alignment mechanism.

For deep speaker verification some simple architectures with only dense layers [4] have been proposed. However, lately it has been tried to employ deep neural networks as Residual CNN Networks [5] but in text-dependent task it has not achieved the same good results as previous simple approaches.

In our network we propose a straightforward architecture with only a few layers which include the use of 1-dimension convolution (1D convolution) layers instead of dense layers or 2D convolution layers as in other works. Our proposal is to operate in the temporal dimension to add context information to the process and at the same time the channels are combined at each layer. The context information which is added depends on the size of the kernel used in convolution layer.

To use this type of layer, it is convenient that the input signals have the same size to concatenate them and pass to the network. For this reason, we apply a transformation to interpolate or fill with zeros the input signals to have all of them with the same dimensions.

The operation of the 1D convolution layers is depicted in Fig. 3, the signal used as layer input and its context, the previous frames and the subsequent frames, are multiplied frame by frame with the corresponding weights. The result of this operation for each frame is linearly combined to create the output signal.

3. Experimental Data

In all the experiments in this paper, we used the RSR2015 text-dependent speaker verification dataset [19]. This dataset consists of recordings from 157 male and 143 female. There are 9 sessions for each speaker pronouncing 30 different phrases. Furthermore, this data is divided into three speaker subset: background (bkg), development (dev) and evaluation (eval). We develop our experiments in Part I of this data set and we employ the bkg and dev data (194 speakers, 94 female/100 male) for training. The evaluation part is used for enrollment and trial evaluation.

4. Results

In our experiments, we do not need the phrase transcription to obtain the corresponding alignment, because one phrase dependent HMM model has been trained with the background partition using a left-to-right model of 40 states for each phrase. With these models we can extract statistics from each utterance of the database and use this alignment information inside our DNN architecture. As input to the DNN, we employ 20 dimensional Mel-Frequency Cepstral Coefficients (MFCC) with their first and second derivatives as features for obtaining a final input dimension of 60. On these input features we apply a data augmentation method called Random Erasing [20], which helps us to avoid overfitting in our models due to lack of data in this database.

On the other hand, the DNN architecture consists of the front-end part in which several different configurations of layers have been tested as we will detail in the experiments, and the second part of the architecture which is an alignment based on HMM models. Finally, we have extracted supervectors as a combination of front-end and alignment with a flatten layer and with them we have obtained speaker verification scores by using a cosine similarity metric without any normalization technique.

A set of experiments was performed using Pytorch [21] to evaluate our system. We compare a front-end with mean reduction with similar philosophy as [4][5] to the feature input directly or a front-end both followed by the HMM alignment. In the part of the front-end, we implemented 3 different layer configurations: one convolutional layer with a kernel of dimension 1 equivalent to a dense layer but keeping the temporal structure and without adding context information, one convolutional layer with a kernel of dimension 3, and three convolutional layers with a kernel of dimension 3.

In Table 1 we show equal error rate (EER) results with the different architectures trained on the background subset for female, male and both partitions together. We have found that, as we expected, the first approach with mean reduction mechanism for extracting embeddings does not perform well for this text-dependent speaker verification task. It seems that this type of embeddings do not represent correctly the information to achieve discrimination between the correct speaker and phrase both simultaneously. Furthermore, we show how changing the typical mean reduction for a new alignment layer inside the DNN achieves a relative improvement of 91.62 % in terms of the EER %.

Nevertheless, these EER results were still quite high, so we decided that the results can be improved training with background and develop subsets together. In Table 2, we can see that if we use more data for training our systems, we achieve better performance especially in deep architectures with more than...
Cuadro 1: Experimental results on RSR2015 part I [19] eval subset, where EER % is shown. These results were obtained by training only with bkg subset.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Fem</th>
<th>Male</th>
<th>Fem+Male</th>
</tr>
</thead>
<tbody>
<tr>
<td>$FE: 3C + \text{mean}$</td>
<td>3 11.20 %</td>
<td>12.13 %</td>
<td>11.70 %</td>
</tr>
<tr>
<td>$Signal + \text{alig.}$</td>
<td>–</td>
<td>1.43 %</td>
<td>1.37 %</td>
</tr>
<tr>
<td>$FE: 1C + \text{alig.}$</td>
<td>1</td>
<td>1.16 %</td>
<td>0.98 %</td>
</tr>
<tr>
<td>$FE: 1C + \text{alig.}$</td>
<td>3</td>
<td>1.01 %</td>
<td>0.77 %</td>
</tr>
<tr>
<td>$FE: 3C + \text{alig.}$</td>
<td>3</td>
<td>0.86 %</td>
<td>1.00 %</td>
</tr>
</tbody>
</table>

One layer, this improvement is observed for both architectures. This fact remarks the importance of having a large amount of data to be able to train deep architectures. In addition, we performed an experiment to illustrate this effect in Fig.4 where we show how if we increase little by little the amount of data used to train, the results progressively improve although we can see that the alignment mechanism makes the system more robust to training data size.

Cuadro 2: Experimental results on RSR2015 part I [19] eval subset, showing EER %. These results were obtained by training with bkg+dev subsets.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Fem</th>
<th>Male</th>
<th>Fem+Male</th>
</tr>
</thead>
<tbody>
<tr>
<td>$FE: 3C + \text{mean}$</td>
<td>3</td>
<td>9.11 %</td>
<td>8.66 %</td>
</tr>
<tr>
<td>$Signal + \text{alig.}$</td>
<td>–</td>
<td>1.43 %</td>
<td>1.37 %</td>
</tr>
<tr>
<td>$FE: 1C + \text{alig.}$</td>
<td>1</td>
<td>1.17 %</td>
<td>0.98 %</td>
</tr>
<tr>
<td>$FE: 1C + \text{alig.}$</td>
<td>3</td>
<td>1.07 %</td>
<td>0.78 %</td>
</tr>
<tr>
<td>$FE: 3C + \text{alig.}$</td>
<td>3</td>
<td>0.58 %</td>
<td>0.70 %</td>
</tr>
</tbody>
</table>

Figura 4: Results of EER % varying train percentage where standard deviation is shown only for both gender independent results.

For illustrative purposes, we also represent our high-dimensional supervectors in a two-dimensional space using t-SNE [22] which preserves distances in a small dimension space. In Fig.5(a), we show this representation for the architecture which uses the mean to extract the embeddings, while in Fig.5(b) we represent the supervectors of our best system. As we can see in the second system the representation is able to cluster the examples from the same person, whereas in the first method is not able to cluster together examples from the same person. On the other hand, in both representations data are auto-organized to show on one side examples from female identities and on the other side examples from male identities.

Furthermore, we illustrate in Fig.6 the same representation in the previous figure, however in this case we represent the embeddings and the supervectors of the thirty phrases from female identities. With this depiction we checked something that we had already observed in the previous verification experiments since the embeddings from mean architecture are not able to separate between same identity with different phrase and same identity with the same phrase which is the base of text-dependent speaker verification task.

Figura 5: Visualizing Mean embeddings vs Supervectors for 1 phrase from male+female using t-SNE, where female is marked by cold color scale and male is marked by hot color scale.

Figura 6: Visualizing Mean embeddings vs Supervectors for 30 phrases from female using t-SNE. Each phrase is marked by one different color scale.

5. Conclusions

In this paper we present a new method to add a new layer as an alignment inside of the DNN architectures for encoding meaningful information from each utterance in a supervector, which allows us to conserve the relevant information that we use to verify the speaker identity and the correspondence with the correct phrase. We have evaluated the models in the text-dependent speaker verification database RSR2015 part I. Results confirm that the alignment as a layer within the architecture of DNN is an interesting line since we have obtained competitive results with a straightforward and simple alignment technique which has a low computational cost, so we can achieve better results with other more powerful techniques.

6. Acknowledgements

This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the project TIN2017-85854-C4-1-R, by Gobierno de Aragon/FEDER (research group T36,17R) and by Nuance Communications, Inc. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.
7. References


Phonetic Variability Influence on Short Utterances in Speaker Verification

Ignacio Viñals, Alfonso Ortega, Antonio Miguel, Eduardo Lleida
ViVoLAB, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain
{ivinalsb,ortega,amiguel,lleida}@unizar.es

Abstract

This work presents an analysis of i-vectors for speaker recognition working with short utterances and methods to alleviate the loss of performance these utterances imply. Our research reveals that this degradation is strongly influenced by the phonetic mismatch between enrollment and test utterances. However, this mismatch is unused in the standard i-vector PLDA framework. It is proposed a metric to measure this phonetic mismatch and a simple yet effective compensation for the standard i-vector PLDA speaker verification system. Our results, carried out in NIST SRE10 text-constant female det. 5, evidence relative improvements up to 6.65% in short utterances, and up to 9.84% in long utterances as well.

Index Terms: Speaker verification, i-vector, short utterance, vocal content.

1. Introduction

Speaker identification is the area of knowledge focused on characterizing a speaker, so any acoustic utterance may be undoubtedly assigned to the active speaker in it. If the range of hypothetical speakers is limited (the target speaker is one out N possible speakers), we refer to the problem as speaker recognition. Otherwise, we normally talk about speaker verification: Given utterances generated by two speakers, enrollment and test, the system must decide whether both speakers are the same or not.

Multiple possibilities have been proposed to best characterize the speaker on an utterance. Some of the first approaches were based on GMMs [1]. These approaches were evolved to subspace techniques such as JFA [2] and the i-vector [3]. Complemented by PLDA [4], i-vectors have become state-of-the-art for the last decade, receiving small modifications in its original formula. Some of them imply the use of DNNs in the form of GMM posteriors [5] or bottlenecks [6]. Only in the last years new approaches only based on DNNs are emerging [7] with the intention of substituting the i-vector as utterance embedding.

Originally designed for long utterances with a unique speaker, the referred techniques applied to short acoustic audios have demonstrated a significant loss of performance. This situation is critical when considering other tasks such as diarization, in which long utterances from a single speaker are uncommon. Hence any improvement in the understanding of short utterances can provide a boost in many speaker related techniques. Multiple works have succeed in mitigating the degradation of short utterances. Some works attempt to work on the extraction models [8]. Others handle the situation by compensating the

This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the 2013 FPI fellowship, the project TIN2017-85854-C4-1-R and Gobierno de Aragón /FEDER (research group T36,17R).

2. Speaker Verification and Short Utterances

Speaker identification is the task focused on the search of patterns to best recognize an individual by means of his voice. Deeply studied for telephone channel, its usefulness has provoked its adaptation to other environments, such as meetings, broadcast, etc.

But, how can we differentiate two speakers by means of their voice, specially when their speech does not have to be the same, i.e. text-independent speaker identification? The answer lies on the different pronunciation of phonemes by two different people. The state-of-the-art, past and present, has been governed by generative models, and specifically the GMM. The GMM-UBM generative model is supposed to represent the average pronunciation for the whole phonetic content. This average pronunciation can assume the role of reference, making possible the estimation of the deviations for each speaker and consequently, speaker identification. This idea has evolved with subspace techniques [2][3], restricting the possible deviations in a limited subspace. Moreover, back-end models such as PLDA [4], have post-processed the obtained deviations, lowering more and more the error rate. Fig 1 illustrates the standard i-vector PLDA framework, which has obtained some of the best results in speaker verification working with long utterances containing a unique speaker.

However, most of these techniques strongly suffer when applied to short utterances. These audios, because of its limited length, do not contain the totality, or at least the majority, of the vocal space. Moreover, the remaining phonemes are poorly represented due to limited information. Unfortunately, state-of-the-art techniques are built to process the totality of the phonetic space assuming the phonetic variability as pronunciation deviations. Consequently, state-of-the-art models invent unreal phonetic content when necessary. Therefore, when comparing short utterances, decisions are made taking into account unreal speech, with no data to back it up. In conclusion, the comparison of short utterances can depend more on the speech rather than the speaker.
3. Phonetic Mismatch Compensation

Short utterances are an already known problem in speaker verification, with several contributions [8][9][10][11][12][13]. Most of these solutions assume a sort of uncertainty term because of the missing information, which must be compensated. This uncertainty term summarizes about how limited is the information in the utterances, but do not pay attention to the detailed missing phonemes. Therefore, this term is used as a sort of quality measure of the utterance representations. Consequently, scores are only compensated by these representation quality approximations, without any concern about the conditional dependencies when comparing enrollment and test. It is not as harmful comparing utterances with similar limited information as with totally mismatched phonetic content.

According to our understanding, the detailed phonetic information is an impressive side information to pay no attention to. Besides its quality is increasing as long as ASR systems evolve. This sort of knowledge allows the identification of the missing acoustic content, making possible some sort of compensation for the missing acoustic content and a fair comparison of short utterances with only the available audio.

Therefore, our proposal is a proof of concept, as a first attempt to include the phonetic information in the evaluation of the trial. In this work we work on the phonetic mismatch between enrollment and test utterances in a speaker verification system. For this reason we have defined a distance between the enrollment and the test utterance for a trial. This distance is defined to measure how different is the acoustic content of enrollment and test utterances, hence measuring how fair the trials are in terms of acoustic similarity. The higher the number of matching phonemes, the more reliable the score is. Similarly, the lower is the phoneme similarity, the less restrictive should the score be, in order to gain robustness against mismatches.

Considering the i-vector PLDA standard framework, we consider the KL distance as the metric between enrollment and test utterances. This metric, is formulated as follows:

\[
KL_{dist}(p, q) = KL(p || q) + KL(q || p) \tag{1}
\]

\[
KL(p || q) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx \tag{2}
\]

where KL represents the Kullback Leibler divergence between distributions \( p \) and \( q \). KL divergence is not symmetric, hence not a distance, so we make use of the symmetric version instead. This distance will compare our phonetic information, in this work the zeroth order Baum-Welch statistics, extracted from the GMM-UBM step in the pipeline. This information, related with the acoustic content in the utterance, has strong relationships with the desired phonetic content. Nevertheless, no side information is required for its extraction.

The obtained distance can be taken into account in any posterior point of the speaker verification system (i-vector extractor, PLDA, etc). In this work, a fusion of the PLDA score with the distance is considered, made by means of logistic regression. The schematic for the tested framework is illustrated in Fig 2.

With this fusion, the new score is able to compensate the phonetic mismatch in the trials, providing at the same time some sort of quality measure. However, this distance just analyzes the
interaction between enrollment and test utterances in the scoring process, but does not analyze the utterance representation itself.

4. Experiments

Our experiments try to analyze the relevance of the phonetic mismatch in short utterances with limited acoustic information. We have opted for a speaker verification task with NIST SRE datasets, with available long utterances (around 5 minutes of speech with a unique speaker). Cohorts from SRE04, SRE05, SRE06 and SRE08 are used to construct the speaker verification system. This system consists of a 2048-Gaussian GMM-UBM and a 400-dimension i-vector extractor. I-vectors are centered, whitened and length-normalized [14] before being evaluated in a 400-dimension PLDA. Gaussianized MFCCs with first and second derivatives are the input for the system.

The described system has evaluated three subsets based on NIST SRE10 det. 5 core-extended core-extended female experiment: The original utterances constitute the reference system with long utterances. Short utterances are created by chopping random segments from the original ones, only reusing that the short utterance contains between 3 and 60 seconds of speech. Another short utterance subset is also extracted, selecting the frames so that the original and the extracted utterances share the same phoneme distribution. This subset is referred as phonetically balanced in the paper. This latest subset is impossible to find in real life, but allows the analysis of short utterances with (short utterances) and without (phonetically balanced) phonetic variability, making comparisons possible.

The first analysis compares the performance of the three subsets, evaluated with the reference speaker verification system. Both sorts of short utterances are expected to yield degraded performance with regard to the original long ones due to the limited information. The relevance of the phonetic variability is checked by direct comparison between these two results. In Table 1 we present the obtained performances for the long original utterances with respect to their shorter versions, either chopped or phonetically balanced.

The results indicate that both types of variability imply a loss of performance, as expected. However, the level of degradation is far from being the same. Whilst the standard chopped short segments obtain 163.69% relative degradation in terms of EER, the phonetically balanced short utterances only gets degraded a relative 26.46%. Therefore, the estimation variability, i.e. how robust is our i-vector due to limited information, is not nearly as influential as the the phonetic variability, present in real life short utterances. It is important to bear in mind that the amount of data evaluated per short utterance is significantly smaller than the original long utterance, sometimes ruling out up to 95% of the original audio. The obtained results for long and short utterances are considered the baseline results for the experiments onwards.

The previous results show a significant impact of the phonetic variability on the utterance modeling capabilities. However, it is still unclear how this variability affects the performance of our system. Hence we have performed a study comparing acoustic mismatch between enrollment and test utterances with the error score. As a first approach, the acoustic mismatch is measured by means of the KL distance between the distributions of the zeroth Baum Welch statistics for both the enrollment and the test utterances. Thus we are comparing which components of the GMM contribute to the i-vector extraction for enrollment and test. The results are exposed in Table 2, comparing the newly proposed distance with the error at the evaluation operating point ($C_{MISS} = 10, C_{FA} = 1, P_{gl} = 0.01$). The results are differentiated between target and non-target populations for a better understanding.

The results indicate that short utterances suffer from the acoustic mismatch between enrollment and test, being much more significant than in long utterances. This extra mismatch occurs with both target and non-target trial populations. However, this extra mismatch does not have the same effect in the error term. Whereas non-target populations are not affected in terms of error, target trials do, explaining the degradation of short utterances. Trials with short utterances fail because the speaker verification system considers the phonetic variability as speaker variability, not differentiating between them.

The proposed solution is the compensation of the original scores by means of the phonetic distance between enrollment and test utterances. As a first approach, we propose a simple yet effective linear regression fusing two systems, the i-vector PLDA and the KL distance. This first approach helps the speaker verification system to notice whether the acoustic mismatch can be degrading the score or not. The results with this score are shown in Table 3.

![Table 1: EER(%) and minDCF metrics for the original long utterances, the chopped short utterance and the phonetically balanced short utterance](image1)

<table>
<thead>
<tr>
<th>Utterance</th>
<th>EER(%)</th>
<th>MinDCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline Long Utterance</td>
<td>3.25</td>
<td>0.16</td>
</tr>
<tr>
<td>Chopped Short Utterance</td>
<td>8.57</td>
<td>0.40</td>
</tr>
<tr>
<td>Phoneme Balanced Short Utterance</td>
<td>4.11</td>
<td>0.20</td>
</tr>
</tbody>
</table>

![Table 2: KL distance and Error (%) for both target and non-target trials depending on the trial length: long utterances (Long), chopped short utterances (Short) and phonetically balanced short utterances (Phon. Balanced). Error estimated at NIST operation point.](image2)

<table>
<thead>
<tr>
<th>Utterance</th>
<th>Long</th>
<th>Short</th>
<th>Phon. Balanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target</td>
<td>1.06</td>
<td>3.62</td>
<td>2.55</td>
</tr>
<tr>
<td>Non-target</td>
<td>1.74</td>
<td>4.61</td>
<td>3.47</td>
</tr>
<tr>
<td>Error (%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target</td>
<td>28.43</td>
<td>80.40</td>
<td>40.79</td>
</tr>
<tr>
<td>Non-target</td>
<td>0.06</td>
<td>0.01</td>
<td>0.03</td>
</tr>
</tbody>
</table>

According to the results, consistent improvements have been obtained, reaching up to 10% relative improvements. Significantly enough, not only short utterances get improved but so do long utterances.

Finally, it is possible to analyze the benefits of the phonetic compensation and its impact with the different populations (target and non-target) in our trial subsets. The comparison between our baseline system and our compensated version is included in Table 4.

The results indicate a significant reduction of the target trials error (False Negative cases) with both short and long utter-
Table 3: EER(%) and MinDCF metrics for trials with original long utterances and short utterances, evaluating with the standard i-vector PLDA system (Baseline) and our proposed compensated version (Compensated)

<table>
<thead>
<tr>
<th>Utterance</th>
<th>EER (%)</th>
<th>MinDCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>3.25</td>
<td>0.15</td>
</tr>
<tr>
<td>Compensated</td>
<td>2.93</td>
<td>0.15</td>
</tr>
<tr>
<td>Short Utterance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>8.57</td>
<td>0.39</td>
</tr>
<tr>
<td>Compensated</td>
<td>8.00</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Table 4: Error (%) in NIST2010 evaluation point estimated for the baseline and the compensated score with both target and non-target trials. The error is expressed for both long utterances (Long) and chopped short utterances (Short)

<table>
<thead>
<tr>
<th>Utterance</th>
<th>Long</th>
<th>Short</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>28.43</td>
<td>80.40</td>
</tr>
<tr>
<td>Non-target</td>
<td>0.06</td>
<td>0.01</td>
</tr>
<tr>
<td>Compensated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target</td>
<td>8.18</td>
<td>32.18</td>
</tr>
<tr>
<td>Non-target</td>
<td>0.76</td>
<td>0.80</td>
</tr>
</tbody>
</table>

ances. Nevertheless, this compensation generates a small increase of False Positive decisions, which were almost null in the baseline situations.

5. Conclusions

In this work we have successfully analyzed the impact of the different variability factors that short utterances have: phonetic and estimation, providing a simple solution to start dealing with them. This solution is able to generate up to 10% relative improvements in terms of error ratios.

We have identified two main sources of variability in short utterances to degrade the performance in speaker verification systems: phonetic variability (what is said) and estimation variability (how reliable is our representation). According to the experiments, both sources of variability imply a decrease of performance, being phonetic variability significantly much more harmful than the estimation one. This is due to the fact that state-of-the-art technologies invent acoustic content when it is missing, a common situation in short utterances.

Besides, our proposed distance metric between utterances has revealed that the loss in performance in short utterances is due to the mismatch in target trials. Phonetic variability is not differentiated from pronunciation variability, sustain of the speaker variability. Therefore, the speaker verification system does not differentiate between speech and speaker mismatch, hence significantly increasing the amount of False Negative evaluated trials.

Finally, our proposal to take advantage of this mismatch distance has obtained limited but consistent improvements. The fusion of the original PLDA log-likelihood ratio score with the KL distance has obtained improvements up to 10% for short and long utterances. This result is specially satisfactory thanks to its simplicity, leaving the remaining framework (i-vector extractor, PLDA, etc.) unaltered. Further work should be done in order to determine best and more efficient ways to make use of this phonetic information.

6. References


Restricted Boltzmann Machine Vectors for Speaker Clustering

Umair Khan, Pooyan Safari and Javier Hernando

TALP Research Center, Department of Signal Theory and Communications
Universitat Politecnica de Catalunya - BarcelonaTech, Spain
{umair.khan, javier.hernando}@upc.edu, pooyan.safari@tsc.upc.edu

Abstract

Restricted Boltzmann Machines (RBMs) have been used both in the front-end and backend of speaker verification systems. In this work, we apply RBMs as a front-end in the context of speaker clustering. Speakers’ utterances are transformed into a vector representation by means of RBMs. These vectors, referred to as RBM vectors, have shown to preserve speaker-specific information and are used for the task of speaker clustering. In this work, we perform the traditional bottom-up Agglomerative Hierarchical Clustering (AHC). Using the RBM vector representation of speakers, the performance of speaker clustering is improved. The evaluation has been performed on the audio recordings of Catalan TV Broadcast shows. The experimental results show that our proposed system outperforms the baseline i-vectors system in terms of Equal Impurity (EI). Using cosine scoring, a relative improvement of 11% and 12% are achieved for average and single linkage clustering algorithms respectively. Using PLDA scoring, the RBM vectors achieve a relative improvement of 11% compared to i-vectors for the single linkage algorithm.

Index Terms: Speaker Clustering, Restricted Boltzmann Machine Adaptation, Agglomerative Hierarchical Clustering.

1. Introduction

In recent years, deep learning architectures have shown their success in various areas of image processing, computer vision, speech recognition, machine translation and natural language processing. This fact has inspired the research community to make use of these techniques in speaker recognition tasks [1, 2, 3, 4, 5]. In speaker recognition tasks, deep learning architectures are used to extract bottle neck (BN) features and to compute GMMs posterior probabilities in a hybrid HMM-DNN model [6, 7].

Generative or unsupervised deep learning architectures like Restricted Boltzmann Machines (RBMs), Deep Belief Networks (DBNs) and Deep Autoencoders have the ability of representational learning power. A first attempt to use RBMs at the backend in a speaker verification task was made in [8]. Efforts have been done by the authors, in order to learn a compact and fixed dimensional speaker representation in the form of speaker vector by using RBMs as a front-end [9]. They also make use of DBNs at the backend in the i-vector framework for speaker verification [10]. As a continuation to these works, a successful attempt was made in our previous work to apply RBMs as a front-end for learning a fixed dimensional speaker representation [11]. This vector representation of speaker was referred to as RBM vector. In [11] it has been shown that the RBM vector preserves speaker specific information and has shown competitive results as compared to the conventional i-vector based speaker verification systems. This has lead us to apply the RBM vector for learning speaker representation in the task of speaker clustering.

Speaker clustering refers to the task of grouping speech segments in order to have segments from same speaker in the same group. Ideally each group or cluster must contains speech segments that belong to the same speaker. On the other hand, utterances from same speakers must not be distributed among multiple clusters. Several approaches to speaker clustering task exist, for example cost optimization, sequential and Agglomerative Hierarchical Clustering [12, 13, 14, 15]. Some approaches rely on commonly used statistical speaker modeling like Gaussian Mixture Models (GMMs) while others use features extracted using Deep Neural Networks (DNNs). For example in [16], BN features extracted from different DNNs are used for speaker clustering. In this work we consider the use of RBMs for vector representation of speakers. We extend the use of RBM vector [11] in the context of speaker clustering. First, we extract RBM vectors for all the speaker utterances in the same way as in [11]. Then, we perform a bottom-up AHC clustering for all the RBM vectors using cosine or Probabilistic Linear Discriminant Analysis (PLDA) scores. We have found that the RBM vector representation of speakers is successful in task of speaker clustering as in speaker verification. The experimental results show that the RBM vector outperforms the conventional i-vectors based speaker clustering using both the cosine and PLDA scoring methods.

The rest of the paper is organized as follows. Section 2 explains the training of a global model referred to as Universal RBM (URBM), RBM adaptation for speaker utterances and RBM vector extraction followed by a Principal Components Analysis (PCA) whitening and dimensionality reduction. Section 3 contains a brief description of the speaker clustering system using RBM vectors. Section 4 is about the experimental setup, database, evaluation metrics and the experiments carried out. In section 5 the obtained results are depicted and finally some conclusions are drawn in section 6.

2. RBM Vector Representation

In this work, we propose the use of a new speaker representation using RBMs in the context of speaker clustering. Fig. 1 shows the basic block diagram of different stages in the RBM vector extraction process and its input to the clustering module. First of all a global model which is referred to as Universal RBM (URBM), is trained using the features extracted from a large amount of background speakers. Then the URBM is adapted to the features extracted from each speaker’s segments that are to be clustered. These models are referred to as adapted RBMs. The visible to hidden connection weights of these adapted models are used to generate the RBM vector for the corresponding
2.1. URBM Training

In order to generate the RBM vector, the first step is to train a global model with a large amount of background data. This is the URBM which is supposed to convey speaker-independent information. The URBM is trained as a single model with the features extracted from all the background speakers. As these features are real valued, we have used Gaussian real-valued units for the visible layer of the RBM. The RBM is trained using the CD-1 algorithm [17] which assumes that the inputs have zero mean and unit variance. Thus the features are Mean Variance Normalized (MVN) prior to the URBM training. Finally, we trained the URBM with a large amount of training samples generated from the background speakers’ features. The URBM is supposed to learn both speaker and session variabilities from the background data [11].

2.2. RBM Adaptation

Once the URBM is trained, we perform speaker adaptation for every speaker’s segment that has to be used for clustering task. The adapted RBM model is trained only with the data of the corresponding speaker’s segment, in order to capture speaker-specific information. During adaptation the RBM model of the speaker segment is initialized with the parameters (weights and biases) of the URBM. This kind of adaptation technique is successfully applied in [18, 19]. The adaptation is also carried out using CD-1 algorithm. In other words, the adaptation step drives the URBM model in a speaker-specific direction. The weight matrix of the adapted models are supposed to convey speaker-specific information of the corresponding speaker. Fig. 2 shows the visualization of the connection weights of the URBM (at the top of the figure) and of two randomly selected speakers (Speaker 1 and Speaker 2, at the bottom of the figure). From the figure, it is clear that the URBM weights are driven in speaker-specific direction which can be discriminative.

2.3. RBM Vector Extraction

After the adaptation step, an RBM model is assigned to each speaker’s segment. We concatenate the visible-hidden connection weights along with the bias vectors of the adapted speaker RBMs in order to generate a higher dimensional speaker vector, referred to as RBM supervector. As in our previous work [11], we apply a PCA whitening with dimensionality reduction to the RBM supervector in order to generate the lower dimensional RBM vector. The PCA whitening transforms the original data to the principal component space. This results in reducing the correlation between the data components. The PCA is trained with the background RBM supervectors and applied to the test RBM supervectors (used in clustering). All the RBM supervectors are mean-normalized prior to applying PCA. In our previous work [11], it has been shown that the RBM vector is successful in learning speaker-specific information in speaker verification task. Thus, we make use of the RBM vector in the context of speaker clustering.

3. Speaker Clustering

We have considered the conventional bottom-up AHC clustering system with the options of single and average linkages. We did not consider the model retraining approach because it is costly in terms of computations as compared to the linkage approaches to clustering [14]. The system starts with initial number of clusters equal to the total number of speaker segments. Iteratively, the segments that are more likely to be from the same speaker are clustered together until a stopping criterion has reached. The stopping criterion can be thresholding the score in order to decide to merge clusters or it can be a
desired (known) number of clusters achieved. The clustering algorithm is based on computing a distance/similarity matrix \( M(X) \) between all the speakers’ segments. Where \( X \) is the set of segments to be clustered. Hence the RBM vectors of all the segments are extracted, the matrix \( M(X) \) is computed by scoring all the RBM vectors against all. Thus for \( N \) RBM vectors, the matrix \( M(X) \) has dimensions \( N \times N \). In every iteration, the segments with minimum/maximum distance/similarity scores are clustered together and the matrix \( M(X) \) is updated. The corresponding rows and columns of the clustered segments are removed from \( M(X) \) and a new row and column are added. The new row and column contains the distance scores between the new and old clusters. The new scores are computed according to the linkage algorithm used. For example segments \( S_a \) and \( S_b \) are clustered in \( S_{ab} \). Then the scores between new cluster \( (S_{ab}) \) and old segment \( (S_n) \) are computed as follows:

(a) Average Linkage:

\[
s(S_{ab}, S_n) = \frac{1}{2} (s(S_a, S_n) + s(S_b, S_n)) \tag{1}
\]

(b) Single Linkage:

\[
s(S_{ab}, S_n) = \max \{s(S_a, S_n), s(S_b, S_n)\} \tag{2}
\]

Where \( s(S_{ab}, S_n) \) is the score between new cluster \( S_{ab} \) and old segment \( S_n \), while \( s(S_a, S_n) \) is the score between old segments \( S_a \) and \( S_n \).

In this way, the process is initiated until a stopping criterion is met. There are two methods to control the iterations: (1) Fix a threshold, and (2) Add an additional information to the system about the desired (known) number of clusters. The system stops when this number is reached. In this work, we did not let the system know any desired number of clusters and we have used the thresholding method. We have tuned a threshold in order to see the performance of the system at different possible working points. The system performance is measured with respect to a ground truth cluster labels. We will discuss evaluation metrics in section 4.

### 4. Experimental Setup and Database

The experiments were performed using the audios from AGORA database, which contains audio recordings of 34 TV shows of Catalan broadcast TV3 [20]. Each show comprises of two parts, i.e., \( a \) and \( b \). So there are 68 audio files in total, of approximate length of 38 minutes each. These files contain segments from 871 adult Catalan and 157 adult Spanish speakers. For the clustering experiments in this work, we have selected 38 audio files for testing and the remaining 30 audio files are used as a background data. The background data is used to train the Universal Background Model (UBM), Total Variability (T) matrix, URBM and PCA. From the testing audio files, we have manually extracted 2631 speaker segments according to ground truth rich transcription. These segments belong to 414 speakers that appears in the audios.

For both the baseline and proposed systems, 20 dimensional Mel-Frequency Cepstral Coefficients (MFCC) features are extracted using a Hamming window of 25 ms with 10 ms shift. For the baseline, a 512 components UBM is trained to extract i-vectors and the PLDA is trained with the background i-vectors, using Alize toolkit [21]. For the proposed system, more than 3000 speaker segments are extracted from the background shows according to the ground truth rich transcription. For each segment, we concatenate the features of 4 neighboring frames in order to generate 80-dimensional feature inputs to the RBMs. With a shift of one frame, we generate almost 10 million samples for the URBM training. The large amount of training samples will favor more efficient learning that will lead to more accurate URBM. All the RBMs used in this paper comprise of 80 visible and 400 hidden units. The URBM was trained for 200 epochs with a learning rate of 0.0005, weight decay of 0.0002 and a batch size of 100. All the adapted RBM models for the test speaker segments are trained with 200 epochs with a learning rate of 0.005, weight decay of 0.000002 and a batch size of 64. The PCA is trained with the background RBM supervectors as discussed in section 2.3. Finally, fixed dimensional RBM vectors are extracted for the speakers’ segments and are used in the speaker clustering experiments. Different dimensions of the RBM vectors are evaluated which will be discussed in the results section.

There are several metrics to measure the performance of speaker clustering. For example cluster impurity (or conversely cluster purity), rand index, normalized mutual information (NMI) and F-measure as described in [22]. We have considered the Cluster Impurity (CI) measure in this work. CI measures the quality of a cluster, to what extent a cluster contains segments from different speakers. However, this metric has a trivial solution when there is only one segment per cluster. To deal with this, Speaker Impurity (SI) is measured at the same time. SI measures to what extent a speaker is distributed among clusters. There is a trade-off between CI and SI [23]. CI and SI are plotted against each other in an Impurity Trade-off (IT) curve and an Equal Impurity (EI) point is marked as working point.

### 5. Results

Different lengths for RBM vectors as well as for i-vectors are evaluated using cosine scoring and average linkage clustering algorithm. The results are shown in the second column of Table 1. From the Table, it can be observed that if the dimension is increased, the performance is improved, both in case of i-vectors and RBM vectors, in terms of Equal Impurity (EI). However, in case of i-vectors, the best choice is 800 dimension. In case of RBM vectors, the 2000 dimensional RBM vectors performs better than the others. In this case, a relative improvement of 11% is achieved compared to 800 dimensional i-vectors. A further increase in the length of RBM vectors beyond 2000, degrades the performance in terms of EI. The third column of Table 1 compares the performance of RBM vector with the baseline i-vectors in case of single linkage algorithm for clustering using

<table>
<thead>
<tr>
<th>Approach</th>
<th>EI% (Cosine Average)</th>
<th>EI% (Cosine Single)</th>
<th>EI% (PLDA Single)</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-vector (400)</td>
<td>49.19</td>
<td>46.26</td>
<td>36.16</td>
</tr>
<tr>
<td>i-vector (800)</td>
<td><strong>46.66</strong></td>
<td><strong>42.19</strong></td>
<td><strong>35.91</strong></td>
</tr>
<tr>
<td>i-vector (2000)</td>
<td>46.79</td>
<td>42.83</td>
<td><strong>35.89</strong></td>
</tr>
<tr>
<td>RBM vector (400)</td>
<td>51.36</td>
<td>39.66</td>
<td>37.36</td>
</tr>
<tr>
<td>RBM vector (800)</td>
<td>47.20</td>
<td>40.02</td>
<td>32.36</td>
</tr>
<tr>
<td>RBM vector (2000)</td>
<td><strong>41.53</strong></td>
<td><strong>37.14</strong></td>
<td><strong>31.68</strong></td>
</tr>
</tbody>
</table>

Table 1: Comparison of speaker clustering results for the proposed RBM vectors with i-vectors. The dimensions of vectors are given in parenthesis. Each column shows Equal Impurity (EI) in % for different scoring and linkage combinations.
cosine scoring. From the table it is seen that, single linkage is a better choice for our experiments. In this case a minimum EI of 37.14% is obtained with 2000 dimensional RBM vectors which has a relative improvement of 12% over 800 dimensional i-vectors. Finally, we evaluated the proposed system using PLDA scoring as well. The PLDA is trained using background RBM vectors for 15 iterations. The number of eigen voices are set to 250, 450 and 500 for RBM vectors of dimensions 400, 800 and 2000 respectively. All the RBM vectors are subjected to length normalization prior to PLDA training. As per the previous results, we performed this experiment with single linkage algorithm only. The results are compared with i-vectors in the fourth column of Table 1. It is observed that 800 and 2000 dimensional RBM vectors has a better EI compared to the respective similar dimensional i-vectors. In this case, the RBM vectors of dimension 2000 has the minimum EI of 31.68% which results in a relative improvement of 11% over the 800 dimensional i-vectors. However, in case of 400 dimensions, the i-vectors outperform RBM vectors.

The Impurity Trade-off (IT) curves for the baseline as well as the proposed system are shown in Figure 3 and 4. In Figure 3 we have shown the evaluation of different dimensions of i-vectors and RBM vectors in the average linkage clustering using cosine scoring. It can be seen that RBM vectors of length 2000 gives better performance than 800 dimensional i-vectors at all working points. On the other hand, RBM vectors of dimensions 400, 800, 2400 and 3000 performs worse than i-vectors. It is observed that 400 and 800 dimensional RBM vectors could not capture enough information about the speaker while 2400 and 3000 dimensional RBM vectors include unnecessary information which degrades the performance. In Figure 4 we have shown a comparison of 2000 dimensional RBM vectors with 800 dimensional i-vectors using both cosine and PLDA scoring with single linkage algorithm for clustering. The choices of dimensions are based on the previous experiments as 2000 dimensional RBM vectors and 800 dimensional i-vectors give the best results with cosine scoring and average linkage. From Figure 4, it can be seen that the RBM vectors performs better at all working points as compared to i-vectors using their respective cosine and PLDA scoring. However, at low Speaker Impurity regions, the RBM vector with cosine scoring outperforms the baseline i-vector with PLDA scoring. In overall, 2000 dimensional RBM vector has a consistent improved performance compared to i-vectors.

6. Conclusions

In this paper we proposed the use of Restricted Boltzmann Machines (RBM)s for speaker clustering task. RBMs are used to learn a fixed dimensional vector representation of speaker referred to as RBM vector. First, a Universal RBM is trained with a large amount of background data. Then an adapted RBM model per test speaker is trained. The visible-hidden weight matrices along with their bias vectors of these adapted RBMs are concatenated to generate RBM supervectors. These RBM supervectors are subjected to a PCA whitening and dimensionality reduction to extract RBM vectors. Two linkage algorithms for Agglomerative Hierarchical Clustering are explored with RBM vectors scored using cosine and PLDA. Using cosine scoring the performance of the proposed system is better for both the linkage algorithms as compared to i-vector based clustering. In overall, single linkage algorithm with 2000 dimensional RBM vectors is the best choice for our experiments, using both cosine and PLDA scoring. We conclude that the RBM vectors can be successfully used as a speaker representation in a speaker clustering task. The best dimension for RBM vectors is found out to be 2000 which gives better performance over i-vectors as well as RBM vectors of other dimensions.
7. References


Speaker Recognition under Stress Conditions

Esther Rituerto-González, Ascensión Gallardo-Antolín, Carmen Peláez-Moreno
Signal Theory and Communications Department
University Carlos III Madrid
erituert@ing.uc3m.es, gallardo@tsc.uc3m.es, carmen@tsc.uc3m.es

Abstract

Speaker Recognition systems exhibit a decrease in performance when the input speech is not in optimal circumstances, for example when the user is under emotional or stress conditions. The objective of this paper is measuring the effects of stress on speech to ultimately try to mitigate its consequences on a speaker recognition task. On this paper, we develop a stress-robust speaker identification system using data selection and augmentation by means of the manipulation of the original speech utterances. An extensive experimentation has been carried out for assessing the effectiveness of the proposed techniques. First, we concluded that the best performance is always obtained when naturally stressed samples are included in the training set, and second, when these are not available, their substitution and augmentation with synthetically generated stress-like samples, improves the performance of the system.

Index Terms: speaker recognition, speaker identification, emotions, stress conditions, data augmentation, synthetic stress

1. Introduction

In recent years the interest to detect and interpret emotions in speech as well as to generate certain emotions in speech synthesis have grown in parallel. It is well-known that speech recognition systems function less efficiently when the speaker is under an emotional state, and in fact, some studies consider emotions in speech as a distortion [1].

To be able to synthesize an emotion in speech, it is necessary to analyze what are the characteristics that make it different from neutral speech. The work done about emotions in speech is very extensive, analysis are performed to study what features or combinations of them carry more information about emotions improving speech recognition rates [2], and some works aim to model emotions in speech by manipulating systematically some of the parameters of human speech, generating synthetic speech that simulates emotions [3].

Moreover, the record-keeping of databases with emotional and neutral speech is difficult as they are either recorded by actors simulating speech under those emotions, or by people under actual emotions, which could be complicated to induce. Nevertheless, stress is not considered a proper emotion, although it is intimately related to anxiety and nervousness, it is a state of mental or emotional tension resulting from adverse or demanding circumstances.

There is plenty of work about the effects of emotions in Automatic Speech Recognition (ASR) or classification of emotions in speech, but there is few work of the effects of emotions in Speaker Recognition (SR), not to mention about stressed speech on SR. Stressed speech is hard to simulate as it appears together with physical changes such as the increase of heart rate and skin perspiration. There are also hardly any databases in which stressed speech is either simulated or recorded under real conditions, along with the difficulty involved in the labelling process.

The research performed on this paper is part of a project called ‘BINDI: Smart solution for Women’s safety XPRIZE’ by UC3M4Safety group [4]. The UC3M4Safety is a multidisciplinary team for detecting, preventing and combating violence against women from a technological point of view. The goal of this project is to develop a wearable solution that will detect a user’s panic, fear and stress through physiological sensor data, speech and audio analysis and machine-learning algorithms. The ability to detect whether the voice belongs to the user or to anyone else, even under stress conditions is where this research comes in.

In this paper we want to analyze how does stress in speech affect speech recognition rates. We aim to find techniques for strengthening speaker recognition systems, either neutralizing the effects of stress or being able to model and synthesize it from neutral speech, to create synthetically stressed speech using data augmentation techniques.

The rest of the paper is organized as follows: in Section 2 we describe the state of the art in speaker recognition and discuss features and classifiers used in literature. In Section 3, we explain the methodology followed for the feature extraction and the data augmentation techniques. Section 4 refers to the experimental set-up and results, and finally in Section 5 we discuss the conclusions and future work.

2. Speaker Recognition Related Work

Speaker Recognition is the automatic detection of a person from the characteristics of their voices (voice biometrics) [5]. We can distinguish two tasks, Speaker Identification and Verification. The first refers to the recognition of a particular user among a known number of users (a multiclass setting), and the second aims at identifying one user versus the rest (binary setting).

2.1. Features

In the literature, many features are usually used for Speaker Recognition, for example: Mel-Frequency Cepstral Coefficients (MFCC) -due to their low complexity and high performance in controlled environments-, Phonetic and Prosodic features [6] or the Linear Prediction coefficients (LP) [7]. All of these features exhibit good performance in the task when used in neutral or emotionless speech.

For speaker recognition under stress conditions, however, there is hardly any previous work, even though, MFCCs, along with Linear Frequency Cepstral Coefficients (LFCC) and Linear Prediction Cepstral Coefficients (LPCC) are cited as important features [8], together with the Pitch, Energy and Duration, which are features that seem to differ between speakers.
2.2. Data augmentation

Data augmentation (DA) is a commonly used strategy adopted to increase the quantity of training data. It is a key ingredient of the state of the art systems for image and speech recognition [9]. It can act as a regularizer in preventing overfitting [10] and improving performance in imbalanced class problems [11], making the whole process more robust and achieving a better performance. It is also very useful for small data sets, as it is our case, to augment the speech database and as a consequence improve accuracy [12].

2.3. Classifiers

Methods such as Gaussian Mixture Models (GMM) are generally used for speaker recognition. Support Vector Machines are widely applied as well [13], [14]. However, several studies suggest the use of Deep Learning for speaker recognition [15] and others prove the improvement in SR performance using Convolutional Neural Networks [16].

In recent years Deep Learning algorithms have skyrocketed in many scientific fields specially when using a large number of data. But for this research, we aim to keep a balance between computational complexity and accuracy, due to the constraints that our targeted device hardware imposes and the reduced amount of data originally available. Also, preliminary tests to compare GMM, SVM and Multi-Layer Perceptron (MLP) led us to choose the later, a precursor of Deep Neutral Networks, due its better performance.

3. Methodology

In this section we describe the theoretical part of the proposed system, the extraction of the features and the data augmentation techniques. The block diagram for the training stage is represented in Figure 1. The test stage is identical with the exception of the Data Augmentation block.

3.1. Feature Extraction

The acoustic features of speech extracted from audio signals should reflect both anatomy (e.g., size and shape of the throat and mouth) and learned behavioral patterns (e.g., voice pitch, speaking style).

We worked with the features extracted in the work done by Alba Mínguez [17] within BINDI for stress detection since the database employed is the same. These are the pitch, first three formants, twelve Mel-Frequency Cepstral Coefficients and the energy of the signals. The short-term features were computed every 10 ms of audio and then a temporal integration was performed over 1s length segments, calculating the mean and standard deviation, and resulting in one feature vector per 1s audio frames, which is the rate at which the accompanying heart rate measures used for labelling stress were taken.

3.2. Data augmentation

As for our device, we would hypothetically have neutral speech for the learning step and we may find stressed speech for testing. For those reasons and regarding to the low number of samples we have, we considered the generation of a synthetically stressed database performing data augmentation for the particular case of stress conditions. To be able to produce stressed speech out of neutral utterances we carried out an analysis, first listening to the audio signals and detecting what differences could be appreciated between them, and second, measuring those differences between stressed and neutral frames for the same speakers.

As a first outcome, we realized that locution speed reflects the stress of a person, we tend to pronounce more words per second and produce longer pauses when stressed. In these same conditions, there is a tendency to rise the frequency of our voices. Thus, the speed and pitch from audio signals are two variables that we aim to modify by using the SOX library [18] in order to artificially simulate speech under stress conditions.

4. Experiments

In this section we present the construction of our system in a block by block basis: we introduce the database, the labelling strategy, the preprocessing of the data, and the experiments carried out.

4.1. Corpus database

We used the so-called VOCE Corpus Database [19], a 45-speaker recordings database in neutral and stress conditions. For each of the users, speech was recorded on 3 different scenarios: recording, prebaseline and baseline, which were acquired respectively, in a public speaking setting where the speaker is supposed to be under stress conditions, the speaker is reading a paper 24 hours before the speech, and again reading the same paper 30 minutes but before the public speaking setting. The heart rate (HR) was also acquired every second for the three recordings.

<table>
<thead>
<tr>
<th>Samples</th>
<th>Neutral</th>
<th>Stressed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>1389</td>
<td>3989</td>
<td>5378</td>
</tr>
<tr>
<td>Set 2</td>
<td>1716</td>
<td>4858</td>
<td>6574</td>
</tr>
<tr>
<td>Total</td>
<td>3105</td>
<td>8847</td>
<td>11952</td>
</tr>
</tbody>
</table>

However we only used 21 speakers out of the 45 due to the lack of properly recorded HR information, noisy audios or absence of recordings. We divided these 21 speakers into two sets, Set 1 was composed of 10 speakers whose HR were coherent with the recordings in the sense that, when a speaker was reading the heart rate remained stable, but on the public speaking setting the HR rose. Set 2 was made out of the other 11 remaining speakers. In Table 1 the number of samples per setting are specified, each sample representing 1s audio frames.

4.2. Preprocessing

For simplicity, we begin with a conversion from stereo to mono of the audio recordings, followed by a downsampling from 44100Hz to 16000Hz to reduce the computational cost of the problem without loosing too much precision. Then, a normalization of the signals in amplitude is achieved to be able to compare between them, and finally the signals go through a voice activity detector (VAD) [20] that removes silent audio frames as those don’t include valuable information to our task.

4.3. Labelling

Labelling an audio signal to determine stress presence is a delicate matter since there is not a prescribed way to do so given stress is non binary and very subjective. Taking a pragmatical perspective, once more we relied on the work done by Alba Mínguez [17] where the recordings of this corpus were labeled...
according to each user’s heart rate (HR). Every 1s audio frame is labelled as stressed or neutral using two different heart rate thresholds. We selected the binarization threshold that gave better results in their report, which was the 75% percentile of the HR of the user.

4.4. Balancing the data

Soon we realized that the data instances were not balanced for each speaker. An adjustment needs to be made for each set and conditions to get consistent estimates as all classes have the same importance. Nevertheless, the use of an over-sampling technique would have a big drawback in our case because some users have significantly more samples than others, and this would create too many artificial samples. To cope with this problem we cropped randomly the neutral samples by a threshold of 120 samples for both sets, and stressed samples over a threshold of 300 samples. Applying an over-sampling technique (in particular, SMOTE) [11] to the new cropped data culminated in new samples resulting in a balanced data set.

4.5. Preliminary Experiments

Originally, for an initial experimental set-up we used the data available for Sets 1 and 2 (21 speakers). This preliminary experiment is made to observe the behaviour of mismatch conditions’ experiments on the speaker recognition rate. First of all, we divided the data in neutral (NS) and stressed speech (S) and experimented training with one type of speech and testing with the other, and then mixing both types. In order to get reliable results, these experiments were repeated 50 times where, in each repetition, at least 50% data was randomly chosen for testing. The results in terms of accuracy (percentage of audio segments correctly classified) are in Table 2.

Table 2: Results for match and mismatch settings

<table>
<thead>
<tr>
<th>Train</th>
<th>Test</th>
<th>Mean (%)</th>
<th>Std (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS</td>
<td>NS</td>
<td>96.73</td>
<td>0.33</td>
</tr>
<tr>
<td>S</td>
<td>NS</td>
<td>79.21</td>
<td>0.90</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>95.87</td>
<td>0.28</td>
</tr>
<tr>
<td>MIX</td>
<td>NS</td>
<td>90.89</td>
<td>0.49</td>
</tr>
<tr>
<td>MIX</td>
<td>MIX</td>
<td>96.05</td>
<td>0.12</td>
</tr>
</tbody>
</table>

As a first conclusion, match settings are beneficial and mismatch are not, as was expected. When training with neutral and testing with stress, accuracy decreases, so it seems that stressed speech does have different characteristics compared to neutral speech. On the contrary, when training with stress and testing with neutral utterances, the decrease in accuracy is not that important, leading us to think that stressed speech could be sparse data in which neutral speech could be contained but not vice versa. About the mixed conditions experiments, the accuracy reached a 96.05%, achieving a good result for this particular task.

4.6. Synthetic Stress

We performed an analysis to measure the differences between the mean pitch from neutral to stressed audio frames for each user using VOICEBOX [20], and we also estimated the average elocution speed for each user. To do this, we obtained an automatic transcription of each of the recordings by using Google Speech Recognition [21] and computed afterwards the mean number of words per second.

The differences of pitch from neutral to stressed speech were between -2% and +7%, increasing an average of 2.2%. As regards to the elocution speed, subjectively, it seems to increase in stressed speech, but our analysis gave us the opposite conclusion. The number of words per second was higher when the user was reading a text, 2.2 words/s in mean, than when the speaker was performing an oral presentation, 1.85 words/s. By listening to the signals, we determined that the words were pronounced faster but there were more pauses in between them, leading to a lower elocution rate in overall.

Thus, we have changed the locution speed and the pitch from the original database, to produce synthetically stressed samples of speech. The pitch was modified in steps of [-6%, -3%, +3%, +6%], and the signals were reproduced at the following speeds [-20%, -15%, -10%, -5%]. All these modifications are applied to the original sets and result in an augmentation of data, one new synthetic set per modification.

As a first conclusion, match settings are beneficial and mismatch are not, as was expected. When training with neutral and testing with stress, accuracy decreases, so it seems that stressed speech does have different characteristics compared to neutral speech. On the contrary, when training with stress and testing with neutral utterances, the decrease in accuracy is not that important, leading us to think that stressed speech could be sparse data in which neutral speech could be contained but not vice versa. About the mixed conditions experiments, the accuracy reached a 96.05%, achieving a good result for this particular task.
After that, we joined Sets 1 and 2, transforming the problem in a 21-speaker SR task, and combined all the synthetic stress data, multiplying by 5 the original dataset. We repeated the 10 experiments 20 times in order to obtain reliable results. The outcome is available in Table 3, and the equivalence between Training Set and Case number is shown in Figure 3.

<table>
<thead>
<tr>
<th>Case</th>
<th>Training data</th>
<th>Set 1 Mean</th>
<th>Set 1 Std</th>
<th>Set 1+2 Mean</th>
<th>Set 1+2 Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NS (70%)</td>
<td>89.71</td>
<td>0.56</td>
<td>78.55</td>
<td>0.6</td>
</tr>
<tr>
<td>2</td>
<td>NS (70%)</td>
<td>98.59</td>
<td>0.16</td>
<td>97.37</td>
<td>0.21</td>
</tr>
<tr>
<td>3</td>
<td>NS (70%)</td>
<td>98.48</td>
<td>0.23</td>
<td>97.21</td>
<td>0.26</td>
</tr>
<tr>
<td>4</td>
<td>NS (70%)</td>
<td>89.97</td>
<td>0.39</td>
<td>80.46</td>
<td>0.53</td>
</tr>
<tr>
<td>5</td>
<td>NS (70%)</td>
<td>99.93</td>
<td>0.05</td>
<td>99.16</td>
<td>0.11</td>
</tr>
<tr>
<td>6</td>
<td>NS (70%)</td>
<td>89.72</td>
<td>0.53</td>
<td>78.19</td>
<td>0.71</td>
</tr>
<tr>
<td>7</td>
<td>NS (70%)</td>
<td>99.88</td>
<td>0.07</td>
<td>99.21</td>
<td>0.13</td>
</tr>
<tr>
<td>8</td>
<td>NS (70%)</td>
<td>99.91</td>
<td>0.07</td>
<td>99.45</td>
<td>0.08</td>
</tr>
<tr>
<td>9</td>
<td>NS (70%)</td>
<td>99.94</td>
<td>0.06</td>
<td>99.22</td>
<td>0.11</td>
</tr>
<tr>
<td>10</td>
<td>NS (70%)</td>
<td>99.91</td>
<td>0.07</td>
<td>98.97</td>
<td>0.14</td>
</tr>
</tbody>
</table>

There are two types of experiments in Table 3: those where we substitute data and the ones where we augment data. As for substituting the original set by a synthetically stressed one, we have experiments 6 and 7, to be compared with experiments 1 and 2 respectively. Data substitution achieves similar results to the experiments with original data when using synthetic data obtained from NS speech for training (case 1 vs. case 6) and better recognition rates when using synthetic data obtained from S speech for training (case 2 vs. case 7).

The data augmentation experiments are 3, 4, 5, 8, 9 and 10. The outcome is indeed positive, the best results are achieved in experiment 8 with a 99.45% of accuracy for Sets 1+2. These results show us that augmenting the data boosts the SR rate.

One of our objectives with these experiments was that experiment 4 could outperform experiment 2, meaning that we accomplished the task of generating appropriate synthetically stressed speech out of neutral. That goal has not been achieved, but at least in Table 3 experiment 4 is better than 6 which, in turn outperforms 1, for Set 1 and for Set 1+2, settling that stressing speech synthetically and using it as training data alongside with the original data, increases the performance of the SR system.

5. Conclusions and future work

In this research our goal was to analyze how stressed speech affects Speaker Recognition systems. We have identified a problem, which is that stressed speech affects negatively when SR systems are trained only with neutral speech.

In the experiments for data substitution, depending on the difference between the synthetic data and the original one, the substitution outperforms the original data. Besides, the modifications over the speed of the audio signals work better for substituting audio utterances than the modifications in pitch.

As regards to the experiments for augmenting the database with artificial stress, we can conclude that the generation of different synthetically stressed utterances of speech and its addition to the database improves substantially the SR results, reaching a 99.45% of accuracy rate in experiment 8 for Sets 1+2.

Due to limited time and computational power, several experiments and methods remained unexplored and left for future work:

- Our target in this research is a Speaker Identification task, a multiclass problem. However, the objective of the device to be built in BINDI is a Speaker Verification system. These two approaches are not straightforwardly comparable but we believe that the problems and solutions can be translated to one another.
- To simulate a real environment in which the recorded voice is not clean, we could add noise to the same database used and analyze its effect.
- Further analyzing the differences between neutral and stressed speech could lead us to finding new modifications to perform to neutral speech to transform it into an appropriate synthetically stressed speech.
- Implementing new methods for recording stressed speech using BINDI such as Stroop Effect games [22] in which the speaker should experiment stress.
- With the use of data augmentation techniques we have collected a much larger database and we could therefore employ more powerful Deep Learning algorithms in the future.

6. Acknowledgements

This work is partially supported by the Spanish Government-MinECo projects TEC2014-53390-P and TEC2017-84395-P.
7. References


Bilingual Prosodic Dataset Compilation for Spoken Language Translation

Alp Öktem¹, Mireia Farrús¹, Antonio Bonafonte²

¹Universitat Pompeu Fabra, Spain
²Universitat Politècnica de Catalunya, Spain

¹{alp.oktem, mireia.farrus}@upf.edu
²antonio.bonafonte@upc.edu

Abstract

This paper builds on a previous methodology that exploits dubbed media material to build prosodically annotated bilingual corpora. The almost fully-automatized process serves for building data for training spoken language models without the need for designing and recording bilingual data. The methodology is put into use by compiling an English-Spanish parallel corpus using a recent TV series. The collected corpus contains 7000 parallel utterances totaling to about 10 hours of data annotated with speaker information, word-alignments and word-level acoustic features. Both the extraction scripts and the dataset are distributed open-source for research purposes.

Index Terms: bilingual corpora, spoken machine translation, prosody

1. Introduction

Recent approaches in speech-to-speech translation research gave focus on the transfer of para-linguistic information between the languages involved. These approaches extend on the classic pipeline of S2S translation, which consists of automatic speech recognition (ASR), machine translation (MT) and text-to-speech synthesis (TTS), and introduces models that connect directly the information in input and target speech signal. Prosody, which is the linguistic information encoded in cues such as stress, intonation and rhythm, directly influences the communicative value of the source utterance and thus needs to be carried to the output to achieve a complete translation. For example, the effect of emphasis transfer in S2S translation is shown to influence directly the quality of translation [1].

Text data to train machine translation models are collected from utterances carrying the same linguistic information in different languages. Same way, data-driven models that deal with prosody transfer necessitate audio data that not only carries the same linguistic information, but also the same para-linguistic content, encoded in each language’s prosody. For example, Anumanchipalli et. al [2] collected 200 parallel sentences from a flight magazine and recorded using a bilingual speaker to achieve intent transfer in S2S translation. Truong et. al [3] recorded 966 parallel segments with acted emphasis in order to achieve emphasis transfer. A fairly recent project SIWIS [4] that focuses on translation of Swiss languages had 171 prompts with emphasis instructions recorded by many speakers as their training data. These and other similar works [5, 6] rely on small data that is created in laboratory conditions and thus partially reflecting naturalness of conversational language.

Our previous work [7] outlined a methodology for compiling such corpora without the need for preparing bilingual prompts and a recording process. Instead, it exploits professionally created bilingual material that is available through dubbed movies. Dubbing is a carefully designed process where the movie content is first translated and then acted by professionals to reflect original movie lines. A methodology that exploits this parallel nature can be used to create bilingual material for any language pair where dubbing is performed. Using this framework, we present an English-Spanish bilingual corpus of 7000 parallel TV series segments annotated with prosodic features to be used in spoken language translation research.

Contributions of this work can be summarized as follows: (1) We cover the shortcomings of the movie2parallelDB framework [8] introduced in [7] by improving word-alignment and parallel segment extraction processes, (2) we expand the methodology to automatically include speaker information from movie scripts, (3) we present an English-Spanish parallel corpus (Heroes Corpus) of 10 hours of audio together with transcriptions and prosodic features available through this link¹.

2. Methodology

The methodology introduced by Öktem et al. in [7] consists of three stages: (1) a monolingual step, where audio+text segments are extracted from the movie in both languages using transcripts and cues in subtitles, (2) prosodic feature annotation and (3) alignment of monolingual material to extract the bilingual segments. They discuss some shortcomings of their methodology. Firstly, they report that the word-aligner tool they use is not precise and fast enough. Secondly, they don’t address the problem of subtitle/audio mismatch in the dubbed language. This problem occurs because translation of the original movie script for subtitling and dubbing are independent processes that require care in different aspects. While text translation for subtitles can be straightforward, dubbed requires the translated utterances to be in sync with the lip movements of the actors. Because of this reason, many times the dubbed sentences differ substantially from subtitles.

Another problem in their approach is in the bilingual

¹http://hdl.handle.net/10230/35572
segment alignment stage. In order to extract the parallel sentences from two languages, they first translate the sentences in one language to the other using a machine translation system. Then, sentences ordered in both languages are matched by selecting most similar pairs of sentences in terms of a translation scoring metric. As sentence structures can differ between two scripts, this approach employing an unreliable MT system can lead to mismatched segments.

This section explains how we addressed these shortcomings and also ensured the following requirements for the segments to be extracted: (1) They should contain the utterance of only one speaker, (2) transcriptions should be exactly what is being spoken in the audio, (3) it should have f0/intensity and speech rate information labeled on word level.

The overall scheme of the corpus extraction methodology is illustrated in Figure 1.

2.1. Audio segment mining using subtitles

Subtitles are the source for obtaining both (1) audio transcriptions, (2) timing information related to utterances in a movie. These information are contained in a standard srt subtitle file, entry by entry like the structure below:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00:00:09,980 --&gt; 00:00:12,256</td>
<td>Please, tell me who I am,</td>
</tr>
<tr>
<td>2</td>
<td>00:00:12,540 --&gt; 00:00:13,974</td>
<td>and what the future holds.</td>
</tr>
<tr>
<td>3</td>
<td>00:00:14,740 --&gt; 00:00:16,572</td>
<td>-We’re in New York.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Where is everyone?</td>
</tr>
</tbody>
</table>

Each subtitle entry is represented by an index, time cues and the script being spoken at that time in the movie. The script portion can consist of multiple (#2), complete (#2,3) or incomplete sentences (#1) and from single (#1,2) or multiple speakers (#3). Using only the time cues for extracting audio segments with complete sentences of a single speaker does not suffice.

For both the objective of obtaining word-level prosodic features and for segmenting multi-speaker portions, we have used a speech-to-text aligner software. Multi-speaker segments were split from the words following speech-dashes. For merging incomplete segments, punctuation information was used.

2.2. Speech-to-text alignment

For speech-to-text alignment, we used the open-source tool Montreal Forced Aligner (MFA) [9]. Forced alignment process is built on an automatic speech recognition system and requires its own acoustic models and a pronunciation dictionary. Although pre-trained models for both English and Spanish is provided through the tool’s website [5], Spanish pronunciation dictionary isn’t openly available. For this reason, we have created a Spanish pronunciation dictionary [4] that uses the same phoneme set as MFA using word list from the open-source spell checker tool ISpell [3] and obtaining their phonetic transcriptions using TransDic [5].

2.3. Speaker annotation through scripts

Movie scripts, which contain dialogue and scene information, are valuable pieces of information for determining the segment speaker labels. Scripts follows approximately the same format: Actor/actress name is followed by the line they say. And in between, there might be non-spoken information in brackets. An example excerpt from a movie script of TV series Heroes is given below:

Claire: What did you do? What the hell is going on?
[Caption: Manhattan 16 years ago]
Noah: (in Japanese) We think she died in the fire.
Claire: Dad? (Hiro covers her mouth to be quiet.)
(Kaito hands baby Claire to Noah.)

Unlike subtitles, scripts don’t have timing information. In order to map subtitle segments with the speaker information we followed an automatic procedure. We first removed all non-spoken text, which is included in brackets. Then, speaker tags and corresponding lines are extracted with regular expressions depending on the format of the script. Next, segments coming from subtitles are mapped one by one to lines in the script. If 70% of the words in a subtitle segment is included in a script turn, then the segment is labeled with the speaker of that turn. We found this metric to successfully label 95% of the segments.

Scripts are usually only available in the original language. The dubbed language segments are labeled the same as their matches after the alignment step.
2.4. Word-level acoustic feature annotation

Each word in the extracted segments is automatically annotated with the following acoustic features: mean fundamental frequency (f0), mean intensity, speech rate and silence intervals (pauses) before and after. The first two features are extracted with the ProsodyTagger toolkit [10] built on Praat [11]. Pause information is calculated from word-boundary information and speech rate is calculated using:

\[
\text{word speech rate} = \frac{\text{#syllables in word}}{\text{word duration}}
\]

To represent speaker independent, perceptual acoustic variations in the segments, both f0 and intensity values are converted into logarithmic semitone scale relative to the speaker norm value. Thus, speaker mean values were represented by zero values in both cases. Semitone values are calculated with the corresponding formula:

\[
\text{semitone}(x, \text{norm}) = 12 \times \log\left(\frac{x}{\text{norm}}\right)
\]

2.5. Cross-lingual segment alignment based on subtitle cues

The first three methodologies presented in this section dealt with extraction of segments in each language. This subsection explains how segments extracted for each language are aligned to create the bilingual segment pairs. We have developed an aligning process based on timing information of the extracted segments. Note that the segment alignments can be one-to-one, one-to-many, many-to-one or many-to-many depending on the sentence structure in the subtitles. To create our own alignment algorithm based on time cues, we first defined a metric that measures the correlation percentage between two sets of ordered segments \(S=(s_1, ..., s_N)\) and \(E=(e_1, ..., e_N)\):

\[
\text{segments correlation} = \max\left(0, \frac{\text{correlating}}{\text{span}} \times 100\right)
\]

\[
\text{correlating} = \min(e_N, s_N) - \max(s_1, e_1)
\]

\[
\text{span} = \max(e_N, s_N) - \min(s_1, e_1)
\]

where \(e_n\) and \(s_n\) denote the starting and ending time of the \(n^{th}\) segment in set \(E\), \(s_1\) and \(e_1\) denote the starting and ending time of the \(x^{th}\) segment in set \(S\).

The alignment procedure is as follows: Two indexes \(i_E, i_S\) are kept which slide through the segments of each language. First, segments corresponding to each index are checked if they correlate more than the \(T_{\text{Sure}}\) threshold. If they do, they are assigned as a one-to-one matched pair. If not, the possibilities of one-to-many, many-to-one or many-to-many matches are considered. This is done through computing the correlations between combinations of the current and two following segments and selecting the most correlating segment set pair. While considering combinations of the segments it is made sure that two merged segments belong to the same speaker and are not more than 10 seconds far from each other.

Although the \(T_{\text{Sure}}\) threshold catches most of the one-to-one mapping segments, we realized that many of them fall below this threshold even if they map. So, we added another decision step that if one-to-one mapping correlation scores higher than merged pairings and it scores above a \(T_{\text{OK}}\) threshold, then it is preferred as a matched pair.

2.6. Output format

We needed to store the corpus segments in a convenient way to use with machine learning based applications. We used the Proscript library [12] for storing the enhanced transcripts. This library makes it possible to store and manipulate speech transcript related data. The segments are stored in csv files that keep the information listed in Table 1. A csv file containing all the segments is created for each episode as well.

Table 1: Segment information kept in a Proscript format csv file.

<table>
<thead>
<tr>
<th>Information</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>word</td>
<td>tokenized</td>
</tr>
<tr>
<td>id</td>
<td>unique word id</td>
</tr>
<tr>
<td>timing</td>
<td>start and end times</td>
</tr>
<tr>
<td>pause</td>
<td>coming before and after</td>
</tr>
<tr>
<td>punctuation</td>
<td>attached to beginning and end</td>
</tr>
<tr>
<td>f0</td>
<td>in Hertz and log-scale (semitones)</td>
</tr>
<tr>
<td>intensity</td>
<td>in Decibels and log-scale</td>
</tr>
<tr>
<td>speech rate</td>
<td>relative to syllables</td>
</tr>
</tbody>
</table>

3. Compiling the Heroes corpus

We put our methodology into practice by compiling a corpus from the science fiction TV series Heroes\(^6\). Originating from United States, Heroes ran in TV channels worldwide between the years 2006 and 2010. The whole series consists of 4 seasons and 77 episodes and is dubbed into many languages including Spanish, Portuguese, French and Catalan. Each episode runs for a length of 42 minutes.

We chose this series as we had access to the DVD’s with Spanish dubbing. Also, we found it to have the Spanish subtitles closest to the Spanish dubbing scripts among other series.

3.1. Raw data acquisition

The DVD’s of the series were obtained from the Pompeu Fabra University Library. Episodes were extracted using the Handbrake software and were saved as Matroska format (mkv) files. Mkv files can hold multiple channels of audios and subtitles embedded in it like DVDs. In order to run our scripts we first needed to extract the audio and subtitle pairs for both languages. Audio is extracted using the mkvextract command line tool\(^7\). As subtitles were embedded as bitmap images in the DVD, we had to

---

\(^6\)Produced by Tailwind Productions, NBC Universal Television Studio (2006-2007) and Universal Media Studios (2007-2010)

\(^7\)https://mkvtoolnix.download/
run optical character recognition (OCR) in order to get srt format subtitles. As OCR is an error-prone process, the resulting srt files needed to be spell checked.

We collected English and Spanish audio of 21 episodes totaling to 25 hours of raw audio and their corresponding subtitles. The episode scripts were obtained from a fan site in the Internet.

### 3.2. Manual subtitle correction work

A speech corpus necessitates properly transcribed speech segments. Our method is based on obtaining transcriptions from subtitles. Although subtitles are highly reliable sources for obtaining proper transcriptions in the original language of the movie, this is not the case in the dubbed languages. This is due to the fact that dubbing transcript needs to satisfy visual alignment such as lip movements, whereas subtitles do not. Also, subtitles are often done in a more concise way to facilitate easy reading. In our case, we observed that the Spanish subtitles were matching the Spanish audio in approximately 80% of the cases. To accommodate this issue, we manually corrected the Spanish subtitles to match with the Spanish audio. Both subtitle transcripts and timestamps had to be corrected. This process was done using a subtitle editing program Aegisub.

An advantage the manual correction process gives is the opportunity to filter out unwanted audio portions that would end up in the corpus. Subtitle segments that contained noise and music, overlapping or unintelligible speech and speech in other languages (e.g. Japanese) were removed during this process. The spell checking and timestamps and script correction of 21 episodes was done by two annotators and took 60 hours in total.

### 3.3. Heroes corpus in numbers

We present the statistics of the first preparation sprint of *The Heroes Corpus*. 21 episodes from season 2 and season 3 were processed. The totaled audio durations of 7000 parallel segments approaches 10 hours (see Table 2). Counts of several linguistic units in the final parallel corpus are presented in Table 3. A summary of how much of the content in one episode ended up in the dataset in average is presented in Table 4.

#### 4. Discussion

The first version of the Heroes corpus shows that our automated method for bilingual corpus building is successful in terms of the quality of the segments extracted. Our manual inspections show that the segments are correctly aligned and the transcripts are correctly stored with the audio segments.

The Spanish subtitle correction task was the only time-consuming part of the whole process. However, it showed that it is also useful for obtaining clean parallel segments. Subtitle segments that were removed during the correction process ensured the elimination of unwanted audio portions.

We can interpret Table 4 to show us the amount of information loss at various stages. The first one being the word-alignment process where in average 5% of the sentences are lost due to the failure of the word aligner in segmenting words. We found out that many of the segments that are lost this way were either noisy or erroneous. The biggest loss happens at the stage of alignment where in average 30% of the segments in each language are left unaligned. This percentage is directly affected by the alignment parameters explained in Section 2.5. For example, selecting a lower $T_{\text{Sure}}$ leads to detecting more aligned segments but also to more mismatches. A similar logic applies to $T_{\text{DK}}$. Also, choosing a lower $T_{\text{Merged}}$ leads to more coverage of the sentences but more as combinations with others, leading to fewer and longer segments. After experimenting with a handful of parameter combinations, we decided on this configuration for obtaining the corpus presented in this paper: $T_{\text{Sure}} = 70\%$, $T_{\text{Merged}} = 80\%$ and $T_{\text{DK}} = 30\%$.

#### 5. Conclusions

We have presented an English-Spanish bilingual corpus of dubbed TV series content. The corpus that consists of 7000 parallel audio segments with transcriptions and annotated prosodic features is made openly available. We hope both our methodology and the corpus we compiled be useful for the speech-to-speech translation research community.

#### 6. Acknowledgements

Special thanks to annotators Sandra Marcos Bonet and Laura Gómez Fisas for their work during the Spanish subtitle correction process. The annotation work carried by the annotators was financed with the 2018 María de Maeztu Reproducibility Award from Department of Information and Communication Technologies of Universitat Pompeu Fabra received by the first author. The second author is funded by the Spanish Ministry through the Ramón y Cajal program.

---

#### Table 2: Corpus duration information.

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total duration</td>
<td>4:45:36</td>
<td>4:43:20</td>
</tr>
<tr>
<td>Avg. duration/segment</td>
<td>0:00:02.44</td>
<td>0:00:02.42</td>
</tr>
</tbody>
</table>

#### Table 3: Word, token, sentence counts and average word count for parallel English and Spanish segments.

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td># words</td>
<td>56320</td>
<td>48593</td>
</tr>
<tr>
<td># tokens</td>
<td>72565</td>
<td>63014</td>
</tr>
<tr>
<td># sentences</td>
<td>9892</td>
<td>9397</td>
</tr>
<tr>
<td>Avg. # words/sentence</td>
<td>5.69</td>
<td>5.17</td>
</tr>
<tr>
<td>Avg. # words/segment</td>
<td>8.04</td>
<td>6.94</td>
</tr>
<tr>
<td>Avg. # sentences/segment</td>
<td>1.41</td>
<td>1.34</td>
</tr>
</tbody>
</table>

---

9https://heroes-transcripts.blogspot.com/  
9http://www.aegisub.org/
7. References


Building an Open Source Automatic Speech Recognition System for Catalan

Baybars Külebi¹, Alp Öktem²,⁴

¹Col·lectivaT SCCL, Barcelona, Spain
²Universitat Pompeu Fabra, Barcelona, Spain

bkulebi@collectivat.cat, alp@collectivat.cat

Abstract

Catalan is recognized as the largest stateless language in Europe hence it is a language well studied in the field of speech, and there exists various solutions for Automatic Speech Recognition (ASR) with large vocabulary. However, unlike many of the official languages of Europe, it neither has an open acoustic corpus sufficiently large for training ASR models, nor openly accessible acoustic models for local task execution and personal use. In order to provide the necessary tools and expertise for the resource limited languages, in this work we discuss the development of a large speech corpus of broadcast media and building of an Catalan ASR system using CMU Sphinx. The resulting models have a WER of 35.2% on a 4 hour test set of similar recordings and a 31.95% on an external 4 hour multi-speaker test set. This rate is further decreased to 11.68% with a task specific language model. 240 hours of broadcast speech data and the resulting models are distributed openly for use.

Index Terms: speech recognition, audio corpus, Catalan, open source

1. Introduction

Development of technology that involves spoken input necessitates three components in order to accomplish the conversion of human speech to machine readable form: (1) Automatic speech recognition (ASR) system, (2) acoustic model and (3) language model. There exists various open source alternatives for ASR systems such as CMU Sphinx, Kaldi and DeepSpeech, however, publicly available acoustic and language models to use with these applications are mainly available for highly-resourced languages. As speech corpora development needed to build the acoustic models is a high-cost process, it is harder to find open source models for lesser-used languages. This puts these languages in major disadvantage in terms of accessibility in technologies with a voice interface.

Catalan with an estimated 9.1 million speakers among 4 different countries¹, is the ninth most spoken language in Europe and second in Spain. There has been development in Catalan speech technology both from inside and outside of Catalonia, and both research and commercial oriented. Research projects that involve Catalan speech recognition include large-vocabulary TECNOPARLA [1] and an earlier telephone conversation targeted project [2], both from Universitat Politècnica de Catalunya (UPC). The speech recognition models resulting from these projects are not available for public use. Only the corpus used to develop the former one is available on demand and is of limited size. On the commercial spectrum, cloud ASR services by companies like Google, Facebook and Speechmatics provide services in Catalan. However, these services are only provided through a centralized server with a fee or a usage limit and does not guarantee data privacy. Also, the fact that they are closed-source clears away any possibility for customization to different needs. Regarding these, although there seems to be reasonable research and development in Catalan ASR systems, there exists no project with an emphasis on free and open sourced access to its resources.

In this work, we present our development of an open source large vocabulary ASR system for Catalan. To build a free and open system, we incorporated a variety of free tools and resources for our use. We chose to use the HMM-based speech recognition system CMU Sphinx principally for its practicality in application development. Although state-of-the-art systems are neural network based, CMU Sphinx is still a popular choice as an ASR toolkit for its computationally low-cost architecture, active community of developers and rich documentation. As for training data we exploited the broadcast media material publicly available through the Catalan public television TV3. Both acoustic and text resources were collected automatically and processed for model training.

With the motivation of making the language accessible to speech technology developers, we have made both the models and the dataset available online. The ready-to-use models, instructions for deployment and a basic script for decoding are distributed in https://github.com/collectivat/cmusphinx-models. 240 hours of broadcast speech data collected during this process is also accessible through this repository.

2. Speech Recognition System

Basic architecture of an ASR system consists of five elements as can be seen in the Figure 1. Before the central decoder can accept speech signals, they need to be converted into a sequence of fixed size acoustic vectors through a process called feature extraction. Later the recognizer makes use of the acoustic model, the language model and the pronunciation dictionary in order to decode the vector sequences into the most likely word sequences they represent. In most cases, a language model consists of N-grams in which the probability of occurrence of a word depends on its N − 1 predecessors and the pronunciation dictionary provides the mapping between each word and its phonetically written form.

The core part of the recognition relies on how the speech is modeled, which depends on how acoustic model parameters are defined and furthermore how they are trained. In many classical ASR systems, Hidden Markov Models (HMMs) are used for modeling the phones (or tri-phones) as finite states each with associated probability distributions. Within the ASR architecture, HMMs are used for evaluating the probability of given speech features (forward-backward algorithm). Furthermore, their use permit the estimation of the the best model parameters for pro-

¹According to the estimate by Secretary of Language Policies of Government of Catalonia
For our task at hand we decided to use the CMU Sphinx ASR system, which has been developed at Carnegie Mellon University over many decades starting from the original Sphinx-I [4] to its more recent and advanced incarnations of Sphinx-3 [5] and Sphinx-4 [6]. For this work, we specifically used the PocketSphinx package within the Sphinx-5prealpha which incorporates algorithms and codes from the previous CMU Sphinx packages [7, 8, 9]. Comparison studies of open-source ASR tools based on training of similar datasets show that Kaldi is far superior in terms of recognition accuracy, mostly due to its inclusion of advanced techniques such as Deep Neural Networks (DNNs) [10]. However a quick look at the standard recipes of Kaldi show that, its training tasks are mostly built to be executed in CPU and GPU clusters whereas CMU Sphinx can train reasonably detailed acoustic models in desktop systems. In addition, although DNN based models are the state-of-the-art, PocketSphinx holds the advantage of being easily deployable in resource limited environments such as hand-held devices. Unfortunately due to its dependence in external libraries such as LAPACK, Kaldi’s usage in hand-held devices end-up being problematic due to memory limitations [11].

3. Data collection and preparation
We have decided to exploit readily available data on the Internet for compiling the necessary training data. We found this approach to be less costly compared to approaches that involve script preparation and recording.

3.1. Raw data collection
In order to gather Catalan audio with transcriptions, we have taken advantage of the fact that Catalan public television makes its programs accessible online for the general public.

<table>
<thead>
<tr>
<th>Speech Corpus</th>
<th>Feature Extraction (Front-end)</th>
<th>Acoustic Models</th>
<th>Language Models</th>
<th>Pronunciation Dictionary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech</td>
<td>Acoustic Models</td>
<td>Language Models</td>
<td>Pronunciation Dictionary</td>
<td></td>
</tr>
<tr>
<td>Audio</td>
<td></td>
<td>Feature Vectors</td>
<td>Graphemes</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: General architecture of an ASR system.

3.2. Acoustic data preparation
For training the acoustic models for an ASR system, audio segments with lengths between 5-20 seconds are needed. The video files we have downloaded have lengths ranging from approximately 7 minutes to 75 minutes. In order to cut these video files to meaningful segments with desired lengths, we have taken advantage of the subtitling.

Before undertaking the segmentation process, as a start, we first had to verify that the subtitle files correspond correctly to the video recordings. In this stage we did not assess the quality of the subtitle cues, but rather checked for more general problems, which we have determined to be three; (1) if the downloaded subtitle file is empty, (2) if the subtitles correspond to a different program, or (3) if the subtitles correspond to the correct video but with wrong timestamps, possibly due to a wrong frame per second assumption made during the export. In order to solve these problems, we first checked whether the subtitle file had cues or not. If it had, we looked at the end time stamp of the last cue and compared it with the duration of the video. In the case of an empty subtitle file, or subtitle cues extending beyond the video duration, we have eliminated the video and subtitle from further processing.

The training segments are defined as word sequences which start and end with silences. We have used the minimum separation duration of 100 ms, and any sequential cues with separations smaller than this amount are grouped together in batches. Our assumption was that longer separations are more likely to signal the beginning and the end of sentences or simply are most like to represent the pauses in speech, hence practically defining the start and the end of the audio segments. After the cues are grouped into batches, their durations are further evaluated and in the case the durations did not fall between 5 and 20 seconds they were eliminated.

Before starting the actual segmentation of the video files, the text content is further filtered. This included getting rid of the explanatory captions; i.e. any content within parenthesis or brackets, and cues starting with #. Additionally, we have eliminated the cues which had content presumably not in Catalan. As a final step we have normalized the text by eliminating the punctuation and symbols. There are exceptions in this stage due to the nature of Catalan morphology, since dashes and apostrophes might be necessary combining pronouns with verbs.

Finally, the complete audio that is extracted from the video file is cut according to this list of the calculated and filtered segments. During the actual segmentation process, we convert the audio files into mono channel with a sampling of 16 kHz.

3.3. Text and language data preparation
Before starting the actual training process using the acoustic data, a phonetic lexicon needs to be built matching each word as they appear in the transcriptions with its phonetic description. This was achieved by using the grapheme to phoneme conversion tool embedded in the rule based speech synthesis tool espeak. To construct a phonetic lexicon, we first extracted all the unique words in the transcriptions, converted them to their IPA (International Phonetic Alphabet) written format and...
Table 1: The programmes used in constructing the ASR system. Table shows their respective themes, total downloaded durations and the final duration used for the training.

<table>
<thead>
<tr>
<th>Programme name</th>
<th>Theme</th>
<th>Downloaded Duration</th>
<th>Filtered Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Art Endins</td>
<td>Outreach</td>
<td>6:22:9</td>
<td>2:04:31</td>
</tr>
<tr>
<td>Quan arribin els marçians</td>
<td>Culture (Current Events)</td>
<td>17:45:38</td>
<td>7:26:05</td>
</tr>
<tr>
<td>Tot un món</td>
<td>Current events</td>
<td>5:20:12</td>
<td>1:39:16</td>
</tr>
<tr>
<td>Valor aefigit</td>
<td>Current events</td>
<td>47:45:34</td>
<td>26:20:37</td>
</tr>
<tr>
<td>Afers Exteriors</td>
<td>Current events, Outreach</td>
<td>53:01:55</td>
<td>30:31:17</td>
</tr>
<tr>
<td>A la presó</td>
<td>Outreach</td>
<td>5:55:23</td>
<td>0:53:01</td>
</tr>
<tr>
<td>Arts i Oficis</td>
<td>Outreach</td>
<td>12:42:31</td>
<td>6:00:14</td>
</tr>
<tr>
<td>Benvinguts a l’hort</td>
<td>Gastronomy, Outreach</td>
<td>11:35:16</td>
<td>6:58:15</td>
</tr>
<tr>
<td>Collita pròpia</td>
<td>Outreach</td>
<td>10:01:30</td>
<td>6:09:03</td>
</tr>
<tr>
<td>Catalunya des del mar</td>
<td>Current events, Outreach</td>
<td>5:40:51</td>
<td>2:07:23</td>
</tr>
<tr>
<td>De llit en llit</td>
<td>Outreach</td>
<td>3:00:33</td>
<td>1:02:05</td>
</tr>
<tr>
<td>Detectiu</td>
<td>Outreach, Fiction</td>
<td>6:47:56</td>
<td>2:22:11</td>
</tr>
<tr>
<td>Economia en colors</td>
<td>Entertainment</td>
<td>16:03:59</td>
<td>7:44:17</td>
</tr>
<tr>
<td>Els dies clau</td>
<td>Outreach</td>
<td>6:19:19</td>
<td>3:23:36</td>
</tr>
<tr>
<td>Fotografies</td>
<td>Outreach</td>
<td>5:31:49</td>
<td>2:29:39</td>
</tr>
<tr>
<td>Generació Selfie</td>
<td>Outreach</td>
<td>6:51:19</td>
<td>2:51:46</td>
</tr>
<tr>
<td>Històries de Catalunya</td>
<td>Outreach</td>
<td>11:28:34</td>
<td>5:37:31</td>
</tr>
<tr>
<td>La salut al cistell</td>
<td>Gastronomy, Outreach</td>
<td>12:04:46</td>
<td>7:49:46</td>
</tr>
<tr>
<td>L’ofici de viure</td>
<td>Outreach</td>
<td>4:19:38</td>
<td>1:26:45</td>
</tr>
<tr>
<td>Millennium</td>
<td>Outreach</td>
<td>29:25:58</td>
<td>16:11:53</td>
</tr>
<tr>
<td>Trinxeres</td>
<td>Outreach</td>
<td>7:15:55</td>
<td>2:37:20</td>
</tr>
<tr>
<td>Programa sindical CCOO</td>
<td>Outreach</td>
<td>12:16:42</td>
<td>7:03:45</td>
</tr>
<tr>
<td>Quèiquécom</td>
<td>Outreach</td>
<td>52:23:31</td>
<td>27:10:41</td>
</tr>
<tr>
<td>Sota terra</td>
<td>Outreach, Entertainment</td>
<td>11:36:03</td>
<td>4:07:41</td>
</tr>
<tr>
<td>Via llibre</td>
<td>Outreach, Entertainment</td>
<td>46:59:25</td>
<td>24:52:51</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>489:57:23</td>
<td>240:15:52</td>
</tr>
</tbody>
</table>

Finally converted these IPA version to the CMU Sphinx readable format. In total we have used 37 phonemes, consistent with the literature on Catalan phonetic corpus [12].

Additional information on the structure of Catalan language is necessary for the decoding phase. Within an ASR system the statistical information on the linguistic grammar and syntax represented through the language model, and these models can be prepared using a sufficiently large text corpus. In this work, we have taken advantage of the subtitle of our audio corpus and merged them with the Catalan OpenSubtitles Corpus [13] to build a basis for our language models.

The final corpus is cleaned from all symbols and punctuation, and numbers are normalized using espeak tool. The complete corpus has 5.3 million tokens with around 100,000 unique tokens which we used for compiling the phonetic lexicon. Using approximately 58k words (which appear at least twice in the corpus) we have prepared one 3-gram (OT_large_3gram) and another 4-gram (OT_large_4gram) language model in ARPA format using the CMU Language Model toolkit (CMUCLMTK).

4. Training

For our training process we have used the standard CMU Sphinx training steps with very minor changes. The training starts with extraction of the Mel-Frequency Cepstral Coefficients (MFCC) between 130 and 6800 Hz; i.e. 12 cepstra using the C0 as the energy component plus their deltas and delta deltas adding up to 39 total parameters (1s_c_d_dd). For the acoustic model training, our Gaussian mixture model contains 32 Gaussian densities, and 6000 tied HMM states.

In our process, we started by estimating the transition probabilities of the Context-Independent (CI) HMMs for forced alignment of the acoustic data. For the forced alignment itself we used the sphinx3_align executable that needed to be compiled apart from the Sphinx-5prealpha library. In this step, the audio files are aligned with their respective transcriptions using the CI models, in the case when there is a mismatch with the transcription and the alignment result, the audio files are eliminated for the following steps. After the non-aligning segments are eliminated we were left with 240 hours of total audio. The final amount of audio per programme used is shown in Table 1.

The transition probabilities of the CI HMMs re-estimated using filtered data set, and following this phase a complete list of tri-phones (58289 in our case) are built and their transition probabilities are estimated in the form of Context-Dependent (CDs) HMMs. These tri-phones account for both between-word and within-word contexts, however since the training data might not account for all the possibilities, the unseen tri-phones are tied to the seen tri-phones using decision trees.

We performed our training in a resource limited environment. For four threads of Intel(R) Atom(TM) CPU N2800 with
1.86GHz, the whole training process took about 120 hours.

4.1. Evaluation
In order to evaluate the word error rate (WER) of the acoustic models we wanted to make sure that the test voices do not appear in the training corpus. In order to evaluate the acoustic models for similar recordings, we downloaded different TV3 programmes that were not used in the training. 4 hours of new TV3 recordings evaluated with the OT_large_4gram language models resulted in a WER of 35.2. For the decoding we used the standard decoding script within the CMU Sphinx.

In order to evaluate the accuracy of the models in a cleaner environment and also to guarantee 100% speaker exclusion we decided to use another test set for evaluation round. For this we used FESTCAT corpus, which is specifically designed for creating a speech synthesis system for Catalan [14, 15], and consists of 28 hours of recordings from 10 different voices (5 female, 5 male). For evaluating our ASR system we used 4 total hours of 4 female and 4 male voices with their corresponding transcripts. It should be noted that due to the clean environment of the recordings, the FESTCAT dataset also represents a more ideal audio quality.

Due to our restricted text corpus, we have created another set of language models in addition to the ones explained in the subsection 3.3 specifically for the test decoding. The second set of language models uses a corpus of FESTCAT text plus the corpus explained in the subsection 3.3. Using this new corpus we created one 3-gram language model with the most frequent 20k words (OT_20k_3gram) and two other models with the most frequent 58k words (OTF_large_3gram, OTF_large_4gram) similar to the OT_large models. Whereas for the OT_large_4gram language model we ended up with a WER of 31.95%, the best results were attained by the OTF_large_4gram model at 11.68%.

The results for each language model with the corresponding real-time decoding factor (xRT) for an Intel i7-4510U 3 Ghz Quad-core architecture is shown in the table 2. The high precision OTF results, show that if the acoustic conditions are perfect and the language models are “in-domain,” our acoustic models can recognize voices that are not in its training set reasonably well. In addition, for our cases the main factor which improved the recognition precision was the amount of pruning that the corpus was subjected to for constructing the language models. Whereas moving from the most frequent 20k words to most frequent 58k words makes a considerable improvement, the effect of using 4-gram instead of 3-gram seems to be very small, probably due to our specific test condition. However one important difference between the 3-gram and 4-gram models is the xRT, for which the 3-gram models are considerably faster than the 4-gram models. Note that we did not undertake any optimization of the decoding parameters neither for best precision nor for the best computational performance.

Table 2: The WER and xRT results for different language models for the FESTCAT test dataset.

<table>
<thead>
<tr>
<th>Language Model</th>
<th>WER (%)</th>
<th>xRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OT_large_4gram</td>
<td>31.95</td>
<td>0.952</td>
</tr>
<tr>
<td>OTF_20k_3gram</td>
<td>22.50</td>
<td>0.872</td>
</tr>
<tr>
<td>OTF_large_3gram</td>
<td>12.11</td>
<td>0.900</td>
</tr>
<tr>
<td>OTF_large_4gram</td>
<td>11.68</td>
<td>1.002</td>
</tr>
</tbody>
</table>

5. Future Work
The most important and basic step for improving our ASR system is to use a better pronunciation dictionary using a better grapheme to phoneme conversion system. In this work, for its ease of use we have taken advantage of espeak, however the festival based FESTCAT speech synthesis system is specifically implemented for Catalan and allows for a more refined grapheme to phone conversion. Training the acoustic model with this improved pronunciation dictionary will allow for better results overall.

For the acoustic data itself depending on the sound quality, background music levels and the speaker mistakes, it should be clustered into clean and other, similar to the librispeech dataset [16]. Additionally, we plan to do a gender diarization model, determining whether the voice is male of female for each segment, in order to assess the gender balance of the whole dataset.

With these acoustic models, it will be possible to do alignment of an audio with its given text. This process will not only be useful in cleaning the dataset itself, but also will allow extending the current set without relying on the cue start and end times within the subtitles. This implies further Catalan acoustic data could be assembled by using the audio and just its corresponding transcription.

Related to this possibility, another tool we would like to develop is a system of automatic punctuation in Catalan. The readability of recognized transcripts depend a lot on sentence segmentation and correct punctuation. The methods for training punctuation engines using recurrent neural networks (RNN) are very well developed, especially with the use of a large text corpus [17]. But also recently it was demonstrated that acoustic data with word-aligned transcriptions can be used to create prosody based punctuation models [18]. For now this type of models have only been trained for English. With the possibility of doing word level alignments for Catalan, we will be able to train one in Catalan in the recent future.

6. Conclusions
In this paper, we have described building of an ASR system for a new language, using only publicly available resources. Applying our methodology on Catalan, we compiled a dataset of 240 hours of transcribed broadcast speech and used it to develop large-vocabulary speech recognition models, both of which are distributed openly online. The accuracy of the resulting models show that they can be a base for speech technology developers to access the Catalan speaking community. Building a voice input interface for a desktop or mobile application is easy as installing the CMU Sphinx toolkit3 and placing the models in its installation directory. The ASR system further gives the possibility to adapt acoustic and language models for more specialized vocabularies and acoustic environments. We believe that the practical and low-cost setup of CMU Sphinx makes it an important player amongst other ASR engines, despite the more modern neural network based alternatives. It keeps its relevance especially for minority languages which have little open acoustic data resources available.

7. Acknowledgements
This project was funded by Softcatala. The authors would like to thank Antonio Bonafonte for his guidance during the writing of this paper.

3https://cmusphinx.github.io/wiki/tutorial/
8. References


Multi-Speaker Neural Vocoder

Oriol Barbany1,2, Antonio Bonafonte1 and Santiago Pascual1
1Universitat Politècnica de Catalunya, Barcelona, Spain
2École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
oriel.barbanymayor@epfl.ch, antonio.bonafonte@upc.edu, santi.pascual@upc.edu

Abstract

Statistical Parametric Speech Synthesis (SPSS) offers more flexibility than unit-selection based speech synthesis, which was the dominant commercial technology during the 2000s decade. However, classical SPSS systems generate speech with lower naturalness than unit-selection methods. Deep learning based SPSS, thanks to recurrent architectures, surpasses classical SPSS limits. These architectures offer high quality speech while preserving the desired flexibility in choosing the parameters such as the speaker, the intonation, etc. This paper exposes two proposals conceived to improve deep learning-based text-to-speech systems. First a baseline model, obtained by adapting SampleRNN, making it as a speaker-independent neural vocoder that generates the speech waveform from acoustic parameters. Then two approaches are proposed to improve the quality, applying speaker dependent normalization of the acoustic features, and the look ahead, consisting on feeding acoustic features of future frames to the network with the aim of better modeling the present waveform and avoiding possible discontinuities. Human listeners prefer the system that combines both techniques, which reaches a rate of 4 in the mean opinion score scale (MOS) with the balanced dataset and outperforms the other models.

Index Terms: deep learning, speech synthesis, recurrent neural networks, text-to-speech, SampleRNN, time series

1. Introduction

Deep learning has revolutionized almost every engineering branch over the latest years, also being successfully applied to text-to-speech (TTS) where it yields state-of-the-art performance and overcomes classical statistical approaches. The time series problem has been completely leveraged by recurrent neural networks (RNNs) and their variants, making them lead to very good results in the speech synthesis field. Moreover, deep generative models can generate speech sample by sample as first proposed in WaveNet [1], which achieved very fine-grained waveform amplitudes and outperformed previous statistical parametric speech synthesis (SPSS) models as a deep auto-regressive system. This paper exposes two of the proposals presented in the bachelor’s thesis of the first author [2] that were applied to better model the speech generated with a multi-speaker deep learning-based model.

In our previous work [3], SampleRNN [4] was adapted to generate coherent speech in Spanish. SampleRNN models the joint distribution of speech samples using a hierarchical RNN, and it can be used to generate speech-like sequences. As the generation is unconditioned (only previous waveform samples are inputs), the generated sequence is a random speech-like waveform with short-time coherent structure that emulates speech but does not resemble proper spoken contents. As in the case of WaveNet [1], we can inject additional controlling features in the input that span long-term structures of the speech, like frames of acoustic information or pitch contours. In [3], a TTS system is presented which consist of two modules. The first one transforms the linguistic features (phoneme, stress, length of sentence, etc.) into an acoustic features sequence. Then this conditions the second module, a SampleRNN, which generates the waveform, hence acting as a neural vocoder. A joint optimization of models improves the generated speech. In this work we focus on the neural vocoder based on SampleRNN. The main motivation is to generalize the system proposed in [3] to many speakers as a shared deep neural network (DNN) structure, as it achieves better results in generating quality speech than learning the parameters of a single isolated speaker [5, 6]. As in our previous work, it transform acoustic features (MFCC, F0, etc.) into speech waveform. However, the speaker code is added as a new feature to control the identity. Our final goal is to control the phonetic content using the acoustic features and the speaker characteristics by means of the speaker id feature. This would allow to generate different synthetic voices using limited labelled data, as the neural vocoder is trained on unlabelled data. It could also be applied to voice conversion [7]. In this case, the input to the vocoder are the acoustic features of one speaker and the speaker id of other speaker. While the capability of changing the speaker identity influences the architecture and the proposals of this work, this paper is focused on how to apply the acoustic features to get the best quality, without changing the speaker identity with respect to the speaker associated with the input acoustic features.

This work proposes two contributions: speaker dependent normalization of the acoustic features, and an acoustic-features look ahead mechanism. The first proposal aims to perform voice conversion. The acoustic features fed to the network and used to condition the generated speech, such as pitch or Mel-frequency cepstral coefficients (MFCC), depend on the speaker, which means that the input to indicate the speaker identity is redundant. This redundancy is identified by the network, which assigns low weights to the speaker identity and make it irrelevant. The speaker-dependent normalization aims to give importance to the speaker identity by isolating the features from the speaker and thus forcing the network to use the speaker identity to generate natural speech for each user.

The look ahead approach questions the causality of time series modeling, which is not needed unless input features are extracted at real time and therefore not known beforehand. In the case of TTS systems, the text which will be uttered is known beforehand and therefore, the acoustic features of all the signal are known or can be predicted before generating the speech waveform. Moreover, in natural speech, the phonemes sound different depending on the context and thus can change depending on future phonemes (co-articulation). By giving information of the future behavior of the predicted sequence, there are
no discontinuities and artifacts can be reduced. This is translated into better quality speech as rated by human listeners.

The proposals were initially conceived to improve the speech obtained with a deep generative network able to model multiple speakers with the same structure. Nevertheless, the speaker-dependent normalization (see section 2.2) could be used as a new pre-processing technique in a variety of problems, and the look ahead approach (section 2.3) can be generalized to time series modeling. Current state-of-the-art TTS models like WaveNet [1], Tacotron [8] or VQ-VAE [9] already model several speakers with a unique model, but do not apply speaker-dependent normalizations, which is shown to deteriorate results. The look ahead proposal is not either mentioned, but it outperformed our baseline model and could be applied to other time series modeling system.

In the next section, first the baseline system is presented. It consists of SampleRNN [4], extended to generate speech conditioned to acoustic features and speaker identity. Then, the speaker dependent normalization and the look ahead are introduced. In section 3, the experimental setup is described. Section 4 presents the experimental results that show how both proposals outperform the baseline system.

2. Multi-speaker Network

2.1. Baseline

The proposed neural vocoders are based on SampleRNN, an unconditional end-to-end neural audio generation model [4] that consists of two recurrent modules running at different clock rates that aim to model the short and long term dependencies of speech signals, and one module with auto-regressive multilayer perceptrons (MLPs) that processes speech sample by sample. The authors of SampleRNN reported that gated recurrent unit (GRU) [10] cells worked slightly better than long short-term memory (LSTM) ones, hence this is the recurrent architecture adopted for this work. The three tier architecture provides flexibility in allocating the amount of computational resources for modeling different levels of abstraction and results very efficient in memory during training. The final output of SampleRNN model is the probability of the current sample value conditioned on all the previous values of the sequence that can be expressed following the chain rule of probability as stated in equation (1). This follows a Multinoulli distribution, which could be unintuitive due to the naturalness of speech signals, which are real-valued, but achieves better results as it does not assume any distribution shape of the data and thus can more easily model arbitrary distributions. In this work, speech samples are quantized with 8 bits, having therefore 256 possible values. Differing from the linear quantization proposed in SampleRNN, we apply a μ—law companding transformation [11] before classifying into the 256 possible classes to flatten the Laplacian-like distribution of the speech signals.

\[
P(X) = \prod_{t=1}^{T} P(x_t| x_{1}, \ldots, x_{t-1}) \tag{1}
\]

In order to generate speech coherent spoken contents, the model was conditioned like in [3] with acoustic features obtained with Ahocoder [12], a high-quality harmonics-plus-noise vocoder that predicts a set of features that can characterize speech signals. The adapted model with its conditioning inputs is depicted in figure 3. This changes the previous equation as a new dependency factor is included, thus new formulation follows equation (2), where \( I_t \in \mathbb{R}^{43} \) stands for a 43-dimensional acoustic vector corresponding to the analysis window of the current sample \( x_t \).

\[
P(X| L) = \prod_{t=1}^{T} P(x_t| x_{1}, \ldots, x_{t-1}, I_t) \tag{2}
\]

Differing from the original SampleRNN model [4] and apart from the previously mentioned addition of the acoustic conditioners that allow to synthesize coherent speech, the authors also incorporated the blocks in the left of figure 3. These aim to differentiate among all the speakers of the database by means of embedding an identifier which is also used to condition the model jointly with the aforementioned Ahocoder features. Hence \( I_t \) is augmented to include the speaker identity by concatenating the embedding to the acoustic features, resulting in a vector \( I_t \in \mathbb{R}^{10} \).

2.2. Speaker-dependent feature normalization

Features fed to a neural network are often previously normalized to control the magnitude of both the activations and gradients in training. With the hypothesis of having speaker-dependent features, an independent normalization for each of the speakers was proposed to isolate the speech features from the source. Maximum and minimum values for each of the parameters were found within the training partition so it could happen that some features of the train or validation partitions overpass the bounds. The chosen normalization function was a simple feature scaling that follow equation (3), which bound each of the features from 0 to 1. This approach could be also applied with other normalization functions like the z—score, i.e. statistical normalization. This last option was not tested before the writing of this paper due to the low improvement in results of this only modification (see table 1). Nevertheless, as it can be seen in the same table, this approach outperforms the other models when combined with the look ahead approach (explained in section 2.3). Therefore, a statistical normalization could also be tested in future work.

\[
\hat{x} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{3}
\]
This proposal aims to give importance to the speaker identity to ideally allow voice conversion without the need of a complex mapping of features. Inspiration came from the behavior of the pitch for every speaker, which is depicted for both speaker-independent and speaker-dependent normalizations in figures 1 and 2 respectively. These plots illustrate the evolution of the logarithmic fundamental frequency for four different speakers including two males and two females that read the exact same text and thus are very similar once normalized following a speaker-dependent approach. Note that there is some time shifting due to different duration of phonemes and pauses but the signal is yet very similar.

After the classical speaker-independent normalization (figure 1), it is very easy to distinguish between females (75, 76) and males (79, 80). This means that it would be impossible to perform voice conversion because the network doesn’t need the speaker identifier for being this information implicit in the features. This is why this redundancy is translated into the futility of this input observed when trying to change the speaker identity at will. The behavior of the pitch once normalized by speaker is very similar if the intonation is comparable. Nevertheless, the other features that are fed to the network (see next section) resulted in very similar normalizations for both speaker-independent and speaker-dependent approaches.

2.3. Look ahead

In the modeling of non-real-time sequences such as the generation of speech in a TTS system, the features that will be fed to the network are known beforehand. This means that, in contrast with a possible phone call where both ends are talking at real time, the features that will condition the sequence at future time steps are always known and thus can be used to better model the generated signal.

With this idea in mind, the causality that speech synthesizers inherited from the vocoders used in decoding is questioned and both the current and future windows of features are fed to the network. This results in a larger model because the number of features is duplicated at each time step but also achieves better quality without the need of more features.

Note that the look ahead approach modifies the architecture because the upper right 1D-convolution block doubles its input size (the original value of 43 is crossed out in the figure and replaced by 86 to accept both the features of the current and future frames).

3. Experimental setup

In this section we characterize the experimental conditions to evaluate the previous approaches. First we describe the speech data used to estimate the models. Then, the acoustic parameters, architecture and learning hyperparameters are outlined. Finally, the methodology used to evaluate the system is described.

3.1. Dataset

The speech dataset used in the experiments is formed by six Spanish voices from the TC-STAR project [13], where half of them are males and the other half are females. The database was unbalanced with one of the female speakers barely having a quarter of speech recording time compared to the others. Notwithstanding some works like [14] recommend balancing the data per user so that all speakers have approximately the same amount of samples to train, we choose to use all the available data per speaker to avoid restricting all of them to only 14 minutes of speech instead of an hour. The total duration of the whole dataset including the six speakers amounts to 5.25 hours, which we divide into 80% for training, 10% for validation and 10% for testing.

3.2. Feature Design and Hyper-parameters

The acoustic parameters are extracted with Ahocoder in frames of length 15 ms shifted every 5 ms, obtaining 40 Mel-frequency cepstral coefficients, the maximum voiced frequency (fv), the logarithmic F0 value and the voiced/unvoiced flag (uv). To tackle the discontinuity in the logF0 statistics in unvoiced signals, the extracted pitch is post-processed with a log-linear interpolation for the unvoiced segments following previous strategies [14].

All these features are thus scaled following either the proposed speaker-dependent or the more classical speaker-independent normalization. The normalized features are then rearranged to match the speech samples dimensions used in training and the speaker embedding is added as an independent input to the system, as mentioned earlier.

The learning strategy was to train each of the models derived from the previous proposals with mini-batch stochastic gradient descent (SGD) using a mini-batch size of 128 and minimizing the negative log-likelihood (NLL). The chosen optimizer is the adaptive moment estimation (ADAM) [15] for its effectiveness in many problems and ease of use. It is an SGD algorithm with adaptive learning rate, having an initial value of $10^{-4}$, which we enhance for our task with an external rate controller known as scheduler. This had two milestones at epochs 15 and 35. In each of these milestones, the learning rate is scaled down by a factor 0.1, which counterattacks the sudden changes in the loss curve that shows up at first epochs. Weight normalization [16] is also used in the 1D-convolutional layers to speed-up the convergence of the model.

3.3. Subjective evaluation

As this is a generative task that involves synthesized nuances in the speech that are difficult to evaluate with any objective metric, a mean opinion score (MOS) test is conducted. The MOS is a rating of the naturalness of the speech signal with an integer scale ranging from 1 to 5. The meaning of each scale value is translated as Excellent (5), Good (4), Fair (3), Poor (2), and Bad (1).

In total 4 systems are evaluated combining both proposed improvements with all possibilities: (1) speaker dependent normalization and look-ahead (Spk-D + LA); (2) speaker independent normalization and look-ahead (Spk-Ind + LA); (3) speaker dependent normalization (Spk-D); and (4) only speaker independent normalization (Spk-Ind). Hence to perform the test 25 subjects were asked to rate each of the 4 proposed systems under a set of 8 test utterances, one per modelled speaker (4 males and 4 females). In total 32 systems were prompted to be rated per listener, and they could listen the different systems as many times as required to compare and rate them. For each sentence, the transcription of the audio was provided to ease the listening, and the audios of each of the different systems synthesizing the same sentence, were disposed side by side to compare, having a random order per utterance (i.e. the system identity was hidden and mixed among the different utterances).
Speaker-dependent normalization achieve substantially better results in naturalness for the generation of speech modeled with balanced datasets when combined with the look ahead approach. Regarding the second proposal, it outperforms the previously achieved scores and reaches state-of-the-art results when combined with the speaker-dependent normalization. Speaker-dependent normalization was not enough for voice conversion purposes, so more complex architectures were proposed in [2]. Nevertheless, human listeners preferred the speech modeled with speaker-dependent normalization and, given the similarity of the features normalized for each speaker, a better quantization could be applied for coding applications or for deployment of neural networks with limited resources.

Whilst the speaker-dependent normalization itself does not seem to improve the results obtained with the classical speaker-independent feature scaling, when combined with the look ahead approach, it achieves the best score with the balanced dataset. In summary, with the combination of the two proposals, a state of the art MOS score has been achieved for a multispeaker speech synthesis system. Both of these approaches were novelties introduced in this work and results show that they can be potentially beneficial to other TTS systems as well as for a bunch of other applications involving features from different sources and modelling of non-real-time sequences.

### 6. Acknowledgements

This research was supported by the project TEC2015-69266-P (MINECO/FEDER, UE).
7. References


Improving the Automatic Speech Recognition through the improvement of Language Models

Andrés Piñeiro-Martín, Carmen Garcia-Mateo, Laura Docio-Fernandez
atlanTTic Research Center, Multimedia Technologies Group, University of Vigo, Spain
apinheiro@gts.uvigo.es, carmen.garcia@uvigo.es, ldocio@gts.uvigo.es

Abstract

Language models are one of the pillars on which the performance of automatic speech recognition systems are based. Statistical language models that use word sequence probabilities (n-grams) are the most common, although deep neural networks are also now beginning to be applied here. This is possible due to the increases in computation power and improvements in algorithms. In this paper, the impact that language models have on the results of recognition is addressed in the following situations: 1) when they are adjusted to the work environment of the final application, and 2) when their complexity grows due to increases in the order of the n-gram models or by the application of deep neural networks. Specifically, an automatic speech recognition system with different language models is applied to audio recordings, these corresponding to three experimental frameworks: formal orality, talk on newscasts, and TED talks in Galician. Experimental results showed that improving the quality of language models yields improvements in recognition performance.

Index Terms: automatic speech recognition, language model, deep neural networks.

1. Introduction

Automatic Speech Recognition (ASR) is essential in applications where interaction with the user is through spoken communication, so that this can remain natural and effective. Such systems base their performance on acoustic models (AM) and language models (LM) that allow for the representation of the statistical properties of speech and language.

These days, with increases in computing power, the use of deep neural networks has spread to many fields. The current ASR systems are predominantly based on acoustic Hybrid Deep Neural Network–Hidden Markov Models (DNN-HMMs) [1] and the n-gram language model [2] [3]. However, in recent years, Neural Network Language Models (NNLM) have begun to be applied [4] [5] [6]. In these, words are embedded in a continuous space, in an attempt to map the semantic and grammatical information present in the training data, and in this way to achieve better generalization than n-gram models.

The arrival of GPUs, multi-core GPUs and increases in computing power have made it possible to apply deep neural networks (DNNs) with multiple hidden layers, which are able to capture high-level, discriminating information about input features. The depth of the created network (the number of hidden layers), together with the ability to model a large number of context-dependent states, results in a reduction in Word Error Rate (WER). The type of neural networks most often used in language modelling are recurrent neural networks (RNNLMs).

The recurrent connections present in these networks allow the modelling of long-range dependencies that improve the results obtained by n-gram models. In more recent work, recurrent networks such as LSTM (Long Short-Term Memory) [7] have also been applied. We should note that models of this type are far more complex than the n-gram model, and hence it takes more time to create them and they require more training data.

Furthermore, these models cannot be used straightforwardly in decoding, due to the large amount of computational resources needed, so the usual approach to their application is to perform a rescoring stage on a previously obtained word lattice using a n-gram language model. There are several algorithms that can implement this rescoring, such as those cited in [8], [9] and [10].

The current paper extends the work described in our previous study [11]. The impact on ASR performance in that study was analyzed when the quality of the text data corpora used for training the language models was increased and improved. The present paper looks specifically at the effect of enlarging the complexity of the models, increasing the order of the statistical models, and applying deep neural networks.

In addition, it should be noted that when working with minority languages such as Galician, obtaining the necessary data to train language models can itself be difficult. Therefore, this study also analyses how to use a modern system when working with languages for which the available data is limited.

The paper is organized as it follows: in Section 2 the ASR system is described. Section 3 then presents the experimental framework. Section 4 reports the experimental results. Section 5 provides a discussion of these, and finally Section 6 offers some final conclusions and suggestions for further lines of research.

2. Description of the ASR system

The ASR system was built using the Kaldi toolkit [12]. The acoustic models use a hybrid DNN-HMM modeling strategy with a neural network based on Dan Povey’s implementation in Kaldi [13]. This implementation uses a multi-spliced TDNN (Time Delay Neural Network) feed-forward architecture to model long-term temporal dependencies and short-term voice characteristics. The inputs to the network are 40 Mel-frequency cepstral coefficients extracted in the Feature Extraction block with a sampling frequency of 16 KHz. In each frame, we aggregate a 100-dimensional iVector to a 40-dimensional MFCC input.

The topology of this network consists of an input layer followed by 5 hidden layers with 1024 neurons with RELU activation function. Asymmetric input contexts were used, with more context in the background, which reduces the latency of the neural network in on-line decoding, and also because it seems to be more efficient from a WER perspective. Asymmetric contexts of 13 frames were used in the past, and 9 frames in the future.

Figure 1 shows the topology used and Table 1 the layerwise context specification corresponding to this TDNN.
Figure 1: TDNN used in acoustics models [13]

Table 1: Context specification of TDNN in Figure 1

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input context</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[-2,+2]</td>
</tr>
<tr>
<td>2</td>
<td>[-1,2]</td>
</tr>
<tr>
<td>3</td>
<td>[-3,3]</td>
</tr>
<tr>
<td>4</td>
<td>[-7,2]</td>
</tr>
<tr>
<td>5</td>
<td>{0}</td>
</tr>
</tbody>
</table>

The network has been trained with material corresponding to TC-STAR [14], with 79 hours of speaking in Spanish, and to Transcrigal [15], with 30 hours of speaking in Galician.

In terms of the language models, when working with n-gram LMs the model was trained using the SRI Language Modeling Toolkit. N-gram models of order 3 and 4 were used, that is, trigrams and tetragrams. A modified Kneser-Ney discounting of Chen and Goodman has also been applied, together with a weight interpolation with lower orders [16].

For training the RNNLMs, the Kaldi RNNLM [12] software was used. The neural network language model is based on a RNN with 5 hidden layers and 800 neurons, where TDNN layers with activation function RELU, and LSTM layers are combined. The training is performed using Stochastic Gradient Descent (SGD), and in several epochs (in our case, 20 epochs). All RNNLM models have been trained with the same material as in the case of n-gram statistical models. Figure 2 shows the topology of the network used.

Figure 2: RNNLM topology used in the ASR system

3. Experimental framework

This section briefly describes both the text corpora used to train the LMs and the testing data used in the experiments.

3.1. Text corpora

Four text corpora were used (more detailed information in [11]):

- **DUVI**: texts from the DU VI (Diario da Universidade de Vigo). This is a small corpus, but it has very clean and representative text, including a large number of current words and expressions.

- **epub library**: a set of 5,000 Spanish novels translated into Galician using an automatic translator [17]. It should be noted that the results obtained using the LM when trained with this text may contain systematic errors due to these Spanish-Galician translations. One of the objectives in creating this corpus arises from the problem of finding large text corpora in minority languages. The results obtained illustrate the impact of using an LM with text translated from a language for which large quantities of text can be found.

- **Ghoxe, Eroski and Vieiros newspapers**: text from the Galicia Hoxe digital newspaper, published in Santiago de Compostela; text from the Eroski news page, which contains news text of a varied nature; and Vieiros, another digital newspaper published entirely in Galician. It is an extensive corpus, with clean and representative material, in that it contains news on a variety of themes.

- **CORGA**: texts from the Corpus de Referencia del Galego Actual (CORGA), composed of different representative texts from books, newspapers, magazines, plays, audiovisual material and blogs. It is a medium-sized corpus with very carefully prepared and representative kinds of texts.

From the above text corpora, four LMs have been trained through a mixture of single models. The mixtures carried out were the following:

- **CLM1**: language model trained with the texts from DUVI, Ghoxe, Eroski and Vieiros newspapers, and CORGA.

- **CLM2**: directly combining the already-trained language models based on DUVI, Ghoxe, Eroski and Vieiros newspapers, and CORGA in a new language model. By doing this, we seek to analyze the differences in results between training new models with the whole text and mixing the already-trained models.

- **CLM3**: combining the single language models based on the epub library and CORGA.

- **CLM4**: combining the single language models based on DUVI, the epub library, Ghoxe, Eroski and Vieiros newspapers, and CORGA.

Table 2 shows the main characteristics of these combined models.

3.2. Testing data

Tests were made on three audio corpora with different characteristics.

1. **First Corpus: Formal Orality**. Corpus with audio recordings of formal orality corresponding to literary
Table 2: Characteristics of the combined Language Models

<table>
<thead>
<tr>
<th></th>
<th>No INV words</th>
<th>Training text size (M of words)</th>
<th>OOV words</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLM1</td>
<td>720,000</td>
<td>118</td>
<td>2.5 %</td>
</tr>
<tr>
<td>CLM2</td>
<td>730,000</td>
<td>118</td>
<td>2.5 %</td>
</tr>
<tr>
<td>CLM3</td>
<td>630,000</td>
<td>352</td>
<td>2.8 %</td>
</tr>
<tr>
<td>CLM4</td>
<td>900,000</td>
<td>435</td>
<td>2.3 %</td>
</tr>
</tbody>
</table>

readings and orally produced and read speeches. It consists of 30 files with an average duration of 3:50 minutes per recording and a total duration of approximately 115 minutes (about 2 hours).

2. Second Corpus: Speech in newscasts. Speech in newscasts. A corpus with audio recordings of television newscasts from TVG (Televisión de Galicia). They present a mixture of spontaneous and planned speech or read speech, but with more contemporary themes and vocabulary than in the first corpus. It consists of 10 files with an average duration of 34 minutes per recording and a total duration of 340 minutes (5 hours and 40 minutes).

3. Third Corpus: Speech in TED Talks. A corpus with audio recordings from TED Talks in Galician [18]. They present planned speech but are not read, being of a spontaneous nature. It consists of 10 files with an average duration of 16 minutes per recording and a total duration of 163 minutes (2 hours and 43 minutes).

4. Experimental results

Three experiments were carried out to assess the impact of the different language models on ASR performance:

- **Experiment 1**: recognition using a single-pass decoding strategy using 3-gram LMs of Table 2.
- **Experiment 2**: rescoring with 4-gram language models.
- **Experiment 3**: rescoring with RNNLMs.

4.1. Experiment 1

The mixture of models described in Section 3.1 has been tested in each of the corpora described in Section 3.2. The results can be seen in Table 3. The SLM column shows the best result obtained by a single LM, that is, without mixing LMs.

Table 3: Combined Language Models Results

<table>
<thead>
<tr>
<th></th>
<th>SLM WER%</th>
<th>CLM1 WER%</th>
<th>CLM2 WER%</th>
<th>CLM3 WER%</th>
<th>CLM4 WER%</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corpus</td>
<td>21.02</td>
<td>17.61</td>
<td>17.51</td>
<td>17.53</td>
<td>18.14</td>
</tr>
<tr>
<td></td>
<td>CI-95%</td>
<td>± 1.84</td>
<td>± 1.76</td>
<td>± 1.71</td>
<td>± 1.78</td>
</tr>
<tr>
<td>Second</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corpus</td>
<td>21.39</td>
<td>21.56</td>
<td>21.52</td>
<td>23.46</td>
<td>22.86</td>
</tr>
<tr>
<td></td>
<td>CI-95%</td>
<td>± 2.63</td>
<td>± 2.58</td>
<td>± 2.55</td>
<td>± 2.78</td>
</tr>
<tr>
<td>Third</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corpus</td>
<td>19.07</td>
<td>18.77</td>
<td>18.68</td>
<td>19.48</td>
<td>19.18</td>
</tr>
<tr>
<td></td>
<td>CI-95%</td>
<td>± 2.24</td>
<td>± 2.44</td>
<td>± 2.57</td>
<td>± 2.70</td>
</tr>
</tbody>
</table>

Table 3 shows that for the first and second corpus it is not possible to reduce the average WER with any of the LM combinations, but it is possible to reduce the confidence interval of the results. Only for the third corpus was the average WER obtained slightly reduced, improving the results of single LMs.

In view of these findings, it can be concluded that model combinations do not significantly reduce the average WER, but do lead to an improvement in the confidence interval. On average these combined models present more robust results than any of the single models.

It is interesting to compare the results obtained by CML1 and CML2. We recall that CML1 is trained by combining all the texts, whereas in CML2 the previously-trained models are mixed. Table 3 shows that the average WER values are lower for CML2, and therefore it is better to combine previously-trained single language models.

4.2. Experiment 2

In this Experiment a tetragram rescoring on the lattice obtained in the previous experiment is performed. Table 4 shows the average WER together with the 95% confidence interval of the rescoring results.

Table 4: Tetragram Language Models Rescoring Results

<table>
<thead>
<tr>
<th></th>
<th>SLM WER%</th>
<th>CLM1 WER%</th>
<th>CLM2 WER%</th>
<th>CLM3 WER%</th>
<th>CLM4 WER%</th>
<th>CI-95%</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>± 1.84 ± 1.76 ± 1.71 ± 1.78 ± 1.77</td>
</tr>
<tr>
<td>Corpus</td>
<td>17.60</td>
<td>17.51</td>
<td>17.52</td>
<td>16.72</td>
<td>17.01</td>
<td>± 1.84 ± 1.75 ± 1.72 ± 1.76 ± 1.79</td>
</tr>
<tr>
<td>Second</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>± 2.63 ± 2.58 ± 2.55 ± 2.78 ± 2.70</td>
</tr>
<tr>
<td>Corpus</td>
<td>22.80</td>
<td>21.46</td>
<td>21.40</td>
<td>22.64</td>
<td>21.79</td>
<td>± 2.63 ± 2.58 ± 2.55 ± 2.78 ± 2.70</td>
</tr>
<tr>
<td>Third</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>± 2.62 ± 2.46 ± 2.56 ± 2.68 ± 2.66</td>
</tr>
<tr>
<td>Corpus</td>
<td>19.45</td>
<td>18.72</td>
<td>18.52</td>
<td>18.45</td>
<td>17.97</td>
<td>± 2.62 ± 2.46 ± 2.56 ± 2.68 ± 2.66</td>
</tr>
</tbody>
</table>

The average WER and the confidence interval for the three corpora analyzed is reduced. In the first corpus the average WER is reduced by approximately 1% (expressed in absolute terms), obtaining a value of 16.72%. The reduction is similar in the second corpus, going from 22.80% to 21.40% of WER. In the third corpus it goes from 19.45% to 17.97%.

It is also interesting to compare the average WER for the three corpora shown in Table 4. The lowest average value is obtained by CLM4, the model that combines the greatest number of single LMs, and the one with the least out of vocabulary (OOV) words. It also shows how all the combinations of models obtain lower WER results than single models, that is, when applying the rescoring of tetragrams, the CLMs are clearly superior, being more robust in confidence interval and with a lower average WER.

4.3. Experiment 3

In this last experiment a rescoring with the RNNLM is applied to the lattices obtained by the decoding of experiment 2, that is, a rescoring RNNLM is applied to the lattice resulting from the rescoring of tetragrams. For this, RNNLMs have been trained using the same text as in the previous experiment. Table 5 shows the results.

Table 5: Combined Language Models and RNNLM Results

<table>
<thead>
<tr>
<th></th>
<th>SLM WER%</th>
<th>CLM1 WER%</th>
<th>CLM2 WER%</th>
<th>CLM3 WER%</th>
<th>CLM4 WER%</th>
<th>RNNLM WER%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>± 1.84 ± 1.76 ± 1.71 ± 1.78 ± 1.77</td>
</tr>
<tr>
<td>First</td>
<td>21.02</td>
<td>17.61</td>
<td>17.51</td>
<td>17.53</td>
<td>18.14</td>
<td>± 1.84 ± 1.76 ± 1.71 ± 1.78 ± 1.77</td>
</tr>
<tr>
<td>Corpus</td>
<td>21.39</td>
<td>21.56</td>
<td>21.52</td>
<td>23.46</td>
<td>22.86</td>
<td>± 2.63 ± 2.58 ± 2.55 ± 2.78 ± 2.70</td>
</tr>
<tr>
<td>Second</td>
<td>19.07</td>
<td>18.77</td>
<td>18.68</td>
<td>19.48</td>
<td>19.18</td>
<td>± 2.24 ± 2.44 ± 2.57 ± 2.70 ± 2.81</td>
</tr>
<tr>
<td>Third</td>
<td>18.92</td>
<td>19.23</td>
<td>19.14</td>
<td>19.27</td>
<td>18.92</td>
<td>± 1.84 ± 1.76 ± 1.71 ± 1.78 ± 1.77</td>
</tr>
</tbody>
</table>

In this experiment a rescoring with the RNNLM is applied to the lattices obtained by the decoding of experiment 2, that is, a rescoring RNNLM is applied to the lattice resulting from the rescoring of tetragrams. For this, RNNLMs have been trained using the same text as in the previous experiment.
Table 5: RNNLM Rescoring Results

<table>
<thead>
<tr>
<th></th>
<th>SLM</th>
<th>CLM1</th>
<th>CLM2</th>
<th>CLM3</th>
<th>CLM4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>First Corpus</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WER%</td>
<td>16.05</td>
<td>16.82</td>
<td>16.63</td>
<td>16.52</td>
<td>16.57</td>
</tr>
<tr>
<td>CI-95%</td>
<td>± 1.69</td>
<td>± 1.74</td>
<td>± 1.76</td>
<td>± 1.77</td>
<td>± 1.80</td>
</tr>
<tr>
<td><strong>Second Corpus</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WER%</td>
<td>22.98</td>
<td>21.61</td>
<td>21.37</td>
<td>23.44</td>
<td>21.39</td>
</tr>
<tr>
<td>CI-95%</td>
<td>± 2.62</td>
<td>± 2.67</td>
<td>± 2.58</td>
<td>± 2.83</td>
<td>± 2.73</td>
</tr>
<tr>
<td><strong>Third Corpus</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WER%</td>
<td>19.27</td>
<td>19.07</td>
<td>18.73</td>
<td>18.82</td>
<td>18.52</td>
</tr>
<tr>
<td>CI-95%</td>
<td>± 2.94</td>
<td>± 3.01</td>
<td>± 2.95</td>
<td>± 3.27</td>
<td>± 2.95</td>
</tr>
<tr>
<td><strong>Average WER in analysis corpora</strong></td>
<td>19.43</td>
<td>19.16</td>
<td>18.91</td>
<td>19.59</td>
<td>18.83</td>
</tr>
</tbody>
</table>

CLM4. We can conclude that the best strategy to reduce WER has been to combine the language models that have provided the best results in the first experiment. The combined models increase vocabulary size, provide more training texts and therefore reduce the OOV words, while also providing a greater robustness against variation in speech.

5. Discussion

This section offers a discussion of the WER results and the use of data in a modern system when working with a minority language.

5.1. WER results

The reduction of the average WER achieved in the first corpus is greater than that achieved in the second and third. Such a difference in behavior between the first corpus and the other two may be due to the different character of the linguistic samples [11]. The first corpus is composed mainly of read texts (written language), while the second corpus presents a high number of speakers, a heterogeneous mixture of speech types (read language, statements by different speakers, situations including noise, music, a mixture of Galician and Spanish, among others).

To see how such a heterogeneous mixture of speech affects the results, all interviews were removed from the recordings, leaving only the speech of presenters and reporters. The results show an absolute reduction of more than 7% compared to the best case for this corpus, obtaining an average WER of 13.88%. Therefore, the speech type of this corpus clearly does affect the results here.

A detailed analysis of the recognition errors can lead to a further reduction of the average WER. Yet it must be taken into account that some of the errors, at least in the oral corpus (not read), must be assumed to be inevitable. These errors reflect the doubts and errors in speakers’ pronunciation, deviations in forms, etc. They might appear as errors in the transcription on which the calculation of the WER is based, but in which the recognition is in fact successful.

Finally, in order to check how far we can get with the current training data, a new RNNLM model was trained, introducing the transcripts of the analyzed corpora into the development text. With this, we sought to model the network so that it was specifically prepared to recognize the corpus on which it was going to be tested. Of course, applying this technique is only possible when the correct transcripts are available. The results show an absolute WER reduction of 0.2%, that is, a small and not significant improvement. This result leads us to conclude that it is difficult to continue reducing the WER with this training data and with these algorithms.

5.2. Use of limited data in a modern system

To train the models that the ASR systems uses, large corpora of audio (for the acoustic model) and text (for the language model) are necessary. In both cases, the type of data that is collected must be representative of the speech that one wants to recognize.

Obtaining a large corpus of audio recordings, with their corresponding transcriptions, and which are representative of the speech, is not easy when working with minority or less well-resourced languages. Large databases with this information are simply not available. Although there may be TV or radio stations that broadcast in the language, it is difficult to obtain such audio material with accurate transcriptions, and they often lack variety in terms of speech types.

This study has shown that one solution to deal with the shortage of resources in acoustic modeling is to use data from languages with similar phonetics. In our case, looking at Galician, the acoustic models have been trained using data from Spanish, multiplying by 4 the amount of information. Spanish, being a widely spoken language, has the necessary resources to obtain correctly transcribed audio corpus. Therefore, in our acoustic modeling, approximately 70% of data has been used in Spanish (over 79 hours of Spanish speaking). The other 30% corresponds to more than 30 hours of Galician.

For language models, a text corpus was created using data obtained from: 1) different magazines and newspapers published in the language; 2) downloading the information present in the Galician version of Wikipedia; 3) information obtained from small text corpora. In order to increase the amount of data available, another solution was to obtain a large corpus of data in another language, and translate it into Galician using a free automatic translation tool.

6. Conclusions

The results obtained for Galician ASR are promising. Improving the training text of the language models and applying RNNLM in decoding resulted in reducing the average WER obtained. However, it has also been shown that increasing the complexity of the system leads to more training data. The strategies applied to work with minority and less well-resourced languages have also contributed to the positive results in recognition.

As a future line of research, we plan to improve the acoustic and language models of the ASR, as well as to use more efficient algorithms in the decoding stage.

7. Acknowledgements

This work has received financial support from the Spanish Ministerio de Economía y Competitividad through project ‘TraceThem’ (TEC2015-65345-P), from the Xunta de Galicia (Agrupación Estratéxica Consolidada de Galicia accreditation 2016-2019, Galician Research Network TecAnDaLi ED431D 2016/011) and the European Union (European Regional Development Fund – ERDF). Our gratitude to the Ramón Piñeiro Institute of the Xunta de Galicia for allowing the use of the CORGA material and for its collaboration in the labeling of the second and third corpora.

8. References

Towards expressive prosody generation in TTS for reading aloud applications

Mónica Domínguez1, Alicia Burga1, Mireia Farrús1, Leo Wanner2,1

1Universitat Pompeu Fabra, Barcelona, Spain
2Catalan Institute for Research and Advanced Studies (ICREA), Barcelona, Spain

Abstract

Conversational interfaces involving text-to-speech (TTS) applications have improved expressiveness and overall naturalness to a reasonable extent in the last decades. Conversational features, such as speech acts, affective states and information structure have been instrumental to derive more expressive prosodic contours. However, synthetic speech is still perceived as monotonous, when a text that lacks those conversational features is read aloud in the interface, i.e. it is fed directly to the TTS application. In this paper, we propose a methodology for pre-processing raw texts before they arrive to the TTS application. The aim is to analyze syntactic and information (or communicative) structure, and then use the high-level linguistic features derived from the analysis to generate more expressive prosody in the synthesized speech. The proposed methodology encompasses a pipeline of four modules: (1) a tokenizer, (2) a syntactic parser, (3) a communicative parser, and (3) an SSML prosody tag converter. The implementation has been tested in an experimental setting for German, using web-retrieved articles.

Index Terms: speech synthesis, thematicity, prosody, communicative structure, information structure.

1. Introduction

The advances in the development of Embodied Conversational Avatars (ECAs) in the last decades have allowed the exploration of increasingly complex communicative strategies in the field of natural language interaction. Contextual information (such as profile information with respect to culture, age and even emotional state) is being used in the process of dialogue management, reasoning and expressive speech generation for the introduction of ECAs in social settings, for instance as social companions for children or elderly, or as health-care assistants [1, 2, 3]. In this context, the expressiveness of the generated speech plays an important role, which tends to be overlooked. Such expressiveness is mostly conveyed by means of prosody and usually involves an intertwined set of linguistic and para-linguistic functions that are difficult to grasp in a computational model.

Recent studies have demonstrated that high-level linguistic structures such as information (or communicative) structure, produced by the natural language generation (NLG) module in the course of sentence or text generation, are instrumental in the derivation of a more varied prosody in text-to-speech (TTS) applications [4]. When such features are not available, synthetic speech is perceived as monotonous, especially in long monologue discourse as observed in reading aloud exercises. In this case, raw text material (for instance, web-retrieved information) is passed as input to the TTS without any contextual or linguistic information.

In what follows, we propose a methodology for enriching web-retrieved texts with prosodic cues in the context of the Knowledge-Based Information Agent with Social Competence and Human Interaction Capabilities (KRISTINA [3]) in its role as reader. The cues are derived from the Information Structure, which is derived automatically for the discourse that is to be read – based on studies which show that when prosody reflects the Information Structure of a sentence, the overall comprehension of the message improves; see, e.g. [5] for German.

In the early 2000ies, there were some attempts to introduce some basic concepts of Information Structure in TTS applications, in particular, thematicity, understood as the partition of a sentence into theme (i.e., what the sentence in about) and rheme (i.e., what is said about the theme); see [6, 7] among others. However, a binary flat representation of thematicity of this kind has been proved to be insufficient to describe long complex sentences, whereas the hierarchical tripartite approach proposed in [8] within the Meaning–Text Theory yields a better correspondence to prosodic patterns as shown in [9, 10].

We introduce an experimental setup that targets reading aloud of a newspaper article in German. The setup is based on the sentence as the linguistic referent unit and on a formal representation of thematicity that follows the principles of the Meaning–Text Theory. The implementation of the proposed thematicity-based prosody enrichment encompasses four modules, one of which is off-the-shelf (i.e., the syntactic parser) and the rest have been developed by the authors: (1) a tokenizer, which prepares the input needed for the parser; (2) a communicative parser, which derives thematicity spans from morpho-syntactic features; and (3) an SSML prosody tag converter, which assigns a variety of SSML prosody tags based on the thematicity structure of each sentence.

The rest of the paper is structured as follows: Section 2 motivates this work, sets the context of the problem and refers to previous work in this area. In Section 3, we present the proposed methodology to automatically derive communicative structure from text and, thereofon, to generate prosodic contours using SSML tags. Section 4 introduces a sample implementation using a female German voice in MaryTTS. The output of this implementation is evaluated by means of a perception test in Section 5. Finally, conclusions are drawn in Section 6.

2. Motivation and Background

Different linguistic schools have long stated that Information Structure (IS), and, in particular, the dichotomy referred to as theme–rheme [11], given–new [12], or topic–focus [13] is related to intonation.1 Moreover, prosody structure on the grounds of thematicity partitions plays a key role in the understanding of a message [14]. Empirical studies in different languages provide evidence that when thematicity and prosody are

1In our work, we use the first denotation, i.e., theme–rheme or thematicity.
appropriately correlated with each other, comprehension of the message is positively affected (cf., e.g., [5] for German and [15] for Catalan). Therefore, there is reason to assume that a conversational application considering the notions of content packaging by means of the relation between thematicity and prosody will benefit from the same advantages as in natural conversation environments. Most of all, conversational avatars in applications for children in educational settings [16], applications for those with special needs [1] as well as for elderly [2] and, in particular, for those with cognitive impairments [17], would greatly benefit from such a communicatively-oriented improvement.

State-of-the-art conversational applications, in particular TTS systems, do not yet include communicative information. The task is not trivial. It involves, in the first place, a communicative theoretical model, automatic tools to parse the Information Structure of a text, and, last but not least, a generative model of the related prosodic contour. Some preliminary attempts to include thematicity in TTS applications were made in the past. Consider, for instance, Steedman’s work [6] on the correlation of theme and rhyme to rising and falling intonation patterns, which was tested in the Festival speech synthesizer [18], and the creation of dedicated tags in MaryTTS [19] for the notions of givenness and contrast [20]. However, these attempts have a major shortcoming in that they use a flat binary thematicity structure, which does not suffice to describe the complexity of content packaging, especially in relation to prosody. Our previous studies (see, e.g. [21]) suggest that hierarchical thematicity based on propositions as described by Mel’čuk [8] constructs a more versatile scaffolding for communicative modeling of computer interaction with humans. As already mentioned above, Mel’čuk’s methodological approach has also demonstrated to be instrumental in natural language generation applications [22, 23].

In contrast to IS models that propose a partition of sentences into a theme and a rhyme, Mel’čuk [8] argues in the context of the Meaning–Text Theory for a tripartite hierarchical division (‘theme’, ‘rhyme’, and ‘specifier’ – the element which sets the utterance’s context) within propositions that further permits embeddedness of communicative spans; consider (1) for illustration of hierarchical thematicity (annotated following the guidelines established in [24]) of the sentence *Ever since, the remaining members have been desperate for the United States to rejoin this dreadful group.*

\[
(1) \text{[Ever since], } \text{SP1 [the remaining members], } \text{T1 [have been desperate for the United States], } \text{T1(R1) [to rejoin this dreadful group], } \text{R1(R1)R1}
\]

A hierarchical thematicity structure of this kind has been shown to correlate better with ToBI labels than binary flat thematicity [9]. Such a correlation still does not solve the problem of a one-to-one mapping between a specific intonation label (e.g., H\textsuperscript{1}) to a static acoustic parameter (e.g., an increase of 50% in fundamental frequency). A more varied range of prosodic cues based on the analysis of the available corpus of read speech annotated with hierarchical thematicity (in the NLG module) and prosody has been proved to yield an improvement in the perception of expressiveness of the synthesized speech [4]. However, there is still no application that can provide an automatic derivation of thematicity-based prosody cues for raw texts that arrive to the TTS application, such as the implementation proposed in this paper.

3. Methodology
This paper proposes an approach that tests the formal representation of information (or communicative) structure proposed by Mel’čuk [8] and its correspondence to prosody in the context of a concept-to-speech (CTS) application, where text coming from a web-retrieved service is input to the TTS engine.

3.1. Objectives
Our work envisages the study of the IS–prosody interface from a methodological perspective based on a speech synthesis implementation setup. The proposed methodology has the following underlying goals:

- to provide automatic tools to investigate the effect of thematicity–prosody correspondence in human-machine interaction contexts;
- to explore the advantages and limitations of a thematicity-based prosody enrichment in speech synthesis;
- to provide a preliminary scaffolding to incrementally add other communicative dimensions, registers and languages;

Such a methodology addresses two main research issues in this field: (i) the lack of implementation settings of the IS–prosody correspondence and (ii) testing of the integration of the IS–prosody interface in computational settings.

3.2. Pipeline
The proposed pipeline sketched in Figure 1 includes four modules:

1. **Tokenizer**: it splits the text into sentences and words. Punctuation marks are also tokenized as required to serve as input for the syntactic parser.

2. **Syntactic parser**: The parser by Bohnet [25, 26] is used. This parser is trained on the TIGER Penn Treebank [27] and outputs a fourteen-columned CONLL file.

3. **Communicative parser**: This rule-based system derives thematicity labels from syntactic structure. It outputs a CONLL file with an added column for communicative structure (i.e., the output CONLL has fifteen columns). For now, it only derives hierarchical thematicity labels.

4. **SSML prosody converter**: It converts the thematicity spans derived by the communicative parser to SSML spans and assigns a variety of prosody tags to each span. This module is based on the tool presented in [28].

The use of the Speech Synthesis Markup Language (SSML) [29] convention for prosody enrichment, as proposed in [28], facilitates the integration of the methodology proposed in this paper within the context of TTS applications.
The correspondence between thematicity and prosody is presented as variations from referent prosody tag values involving fundamental frequency (F0) and speech rate (SR) over thematicity spans (cf. Table 1). We propose testing a varied range of values generated automatically, against a manual implementation following the findings in [10, 4], where a variety of prosodic cues for each thematicity span is presented based on corpus analysis.

<table>
<thead>
<tr>
<th></th>
<th>F0</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>+15%</td>
<td>-15%</td>
</tr>
<tr>
<td>R1</td>
<td>+10%</td>
<td>+10%</td>
</tr>
<tr>
<td>SP1</td>
<td>+20%</td>
<td>-10%</td>
</tr>
<tr>
<td>P</td>
<td>+15%</td>
<td>-10%</td>
</tr>
</tbody>
</table>

Table 1: Referent prosody tag values for LI thematicity.

Table 1 shows the referent modification for theme (T1), rheme (R1) and specifier (SP1) spans within LI thematicity. Propositions are defined as clauses that contain a finite verb and they are the referent units for thematicity segmentation. They can include LI and L2 spans and embrace under one communicative label different types of syntactic relationships, for example coordination, juxtaposition and subordination. The referent values assigned to each span are chosen randomly within a range of plus minus 5 points in each new sentence. Thus, even though the annotation of thematicity in this experiment is restricted to the sentence domain, an automatic variation is envisaged to generate a different range of prosodic parameters across sentences.

4. Experimental Setup

A working corpus has been created from web-retrieved text in German on advice for sleeping routines and local news. The corpus contains eight texts with a total of 1,418 words. In what follows, we present the experimental setup with respect to the prosody enrichment procedure and the IS–prosody correspondence.

The open source software MaryTTS\(^5\) [19] was used for the implementation. The default synthesized speech output has been enriched using MaryXML prosody specifications\(^6\), which follow the SSML recommendation\(^7\).

The SSML prosody tags allow control of six optional attributes (overall pitch, pitch contour, pitch range, speech rate, duration and volume). These attributes can be modified independently or in combination. For our implementation, overall pitch and speech rate were chosen individually and in combination. Absolute (e.g., ‘+50Hz’ for increasing a specific amount of hertz (Hz) in F0) and relative values can be used to apply the modification. An example of a SSML prosody tag for modification of two prosodic elements is presented below:

Example (1)  
<prosody rate="-10%" pitch="+20%">text to be modified</prosody>

Moreover, the SSML boundary tag that controls the introduction of pauses at a specific location was also used after each thematicity span. The duration of the break is specified in milliseconds (ms). An example of SSML boundary tag is introduced below:

Example (2)  
Text before the break <boundary duration="100"/>text after the break.

The correspondence between thematicity and prosody is presented as variations from referent prosody tag values involving fundamental frequency (F0) and speech rate (SR) over thematicity spans (cf. Table 1). We propose testing a varied range of values generated automatically, against a manual implementation following the findings in [10, 4], where a variety of prosodic cues for each thematicity span is presented based on corpus analysis.

\(^4\)The code of both the communicative parser and the thematicity to SSML module is available in the following repository under a GNU v.3 licence: https://github.com/TalnUPF/KRISThem2ProsModule

\(^5\)Available at http://mary.dfki.de/

\(^6\)http://mary.dfki.de/documentation/maryxml/index.html

\(^7\)https://www.w3.org/TR/speech-synthesis/
5. Evaluation

For the evaluation of automatic assignment of thematicity-based prosody, a selection of newspaper articles in German has been done. From those articles, a selection of sentences with different communicative structures has been made for the perception test, as detailed below.

For the evaluation of the thematicity-based prosody enrichment module, expressiveness was assessed by means of a perception test using: (1) a Mean Opinion Score (MOS) with a 5-point Likert scale: 1-bad, 2-poor, 3-fair, 4-good, and 5-excellent; and a pairwise comparison. Seventeen participants took part in the evaluation, all of them either native speakers of German or proficient speakers. The test was conducted fully in German and participants were informed that our goal is to investigate if synthesized speech was perceived as better expressing the communicative content of the sentence taking into account prosodic variability. Six sentences were included in the perception test representative of different complexity in syntax and communicative structure:

- S1 Warne Fuß und Vollbäder direkt vor dem Schlafengehen fördern den Nachtschlaf.
- S2 Der Begriff der Schlafhygiene bezeichnet Verhaltensweisen, die einen gesunden Schlaf fördern.
- S3 Dafür sorgen, dass der Schlafzimmer ruhig und dunkel ist und eine angenehme Temperatur hat.
- S4 Landrat Thomas Reumann schlägt vor, den Finanzierungsantrag zu stellen, will aber erst im Haushalt 2018 Gelder einstellen.
- S5 Das funktioniert nur, wenn alle mitmachen.
- S6 Im übrigen betonte er, dass der Landkreis nicht allein sei, sondern Städte und Gemeinden als Partner habe, die den Beschluss mittragen müssten.

Three samples of each sentence were included in the MOS test: (1) the default TTS output (DEF); (2) automatic thematicity-based modifications (AUT) and (3) manual thematicity-based prosody modifications (MAN). The pairwise comparison included the default TTS output versus the automatic thematicity-based prosody modification. A total of fifty-one answers are considered in the evaluation. Table 2 shows results of the MOS test. In all cases, the best scoring sample is the thematicity-based prosody modification (either manual or automatic). This supports the initial hypothesis that thematicity-based prosody modifications are perceived as more expressive. In sentences 2, 3, 5 and 6 the best scoring option is the automatic version, whereas sentences 1 and 4 score best for the manual version of the modification. These results are in line with the pairwise comparison shown in Table 3, where all choices go for the thematicity-based modification except for sentence 4.

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEF</td>
<td>2.65</td>
<td>3.35</td>
<td>3.12</td>
<td>2.71</td>
<td>3.18</td>
<td>3.06</td>
<td>3.01</td>
</tr>
<tr>
<td>AUT</td>
<td>3.00</td>
<td>3.53</td>
<td>3.41</td>
<td>2.65</td>
<td>3.53</td>
<td>3.71</td>
<td>3.30</td>
</tr>
<tr>
<td>MAN</td>
<td>3.12</td>
<td>2.88</td>
<td>3.06</td>
<td>3.00</td>
<td>2.71</td>
<td>2.88</td>
<td>2.94</td>
</tr>
</tbody>
</table>

Table 2: Results from the MOS test for the thematicity-based prosody enrichment module

A t-test shows that overall (average) results for the automatic prosody modifications (AUT = 3.30) achieve statistical significance at p < 0.05 compared to the default score (DEF = 3.01).

6. Conclusions

Given the relevant role of the Information Structure–prosody interface in human communication, it seems reasonable that next generation conversational agents face new challenges in adopting communicatively-oriented models. Current speech technologies have been oblivious to advances in theoretical fields studying this correlation, basically due to the lack of a formal representation of the communicative (or information) structure and limited capabilities of prosody enrichment standards to achieve variability in implementation settings.

The present study provides a methodology for a more versatile integration of the IS–prosody interface in TTS for reading aloud applications. Such a methodology contributes in several aspects to the state of the art: (i) a formal description of hierarchical thematicity is used; (ii) a communicative parser that derives thematicity labels is introduced; and (iii) the prosodic cues are automatically derived and tested in a TTS application. All in all, this study pivots the transition from theoretical work on the IS–prosody interface to the integration of thematicity-based prosody enrichment to achieve more expressive synthesized speech.

A limitation of the current study is that it only considers relative acoustic parameters over rather large text segments. Key aspects of prosody modeling, like F0 contour generation in terms of prominence and phrasing remain to be looked into. Future work is, furthermore, aimed at exploring other dimensions of communicative structure like emphasis and foregroundedness within the framework that has been proposed in this paper.

7. Acknowledgements

This work is part of the KRISTINA project, which has received funding from the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement number H2020-RIA-645012. It has been also partly supported by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502). The third author is partially funded by the Ramón y Cajal program.
8. References


Performance evaluation of front- and back-end techniques for ASV spoofing detection systems based on deep features

Alejandro Gomez-Alanis\textsuperscript{1}, Antonio M. Peinado\textsuperscript{1}, Jose A. Gonzalez\textsuperscript{2}, and Angel M. Gomez\textsuperscript{1}

\textsuperscript{1}University of Granada, Granada, Spain
\textsuperscript{2}University of Malaga, Malaga, Spain

\{agomezalanis, amp, amgg\}@ugr.es, j.gonzalez@uma.es

Abstract

As Automatic Speaker Verification (ASV) becomes more popular, so do the ways impostors can use to gain illegal access to speech-based biometric systems. For instance, impostors can use Text-to-Speech (TTS) and Voice Conversion (VC) techniques to generate speech acoustics resembling the voice of a genuine user and, hence, gain fraudulent access to the system. To prevent this, a number of anti-spoofing countermeasures have been developed for detecting these high technology attacks. However, the detection of previously unforeseen spoofing attacks remains challenging. To address this issue, in this work we perform an extensive empirical investigation on the speech features and back-end classifiers providing the best overall performance for an anti-spoofing system based on a deep learning framework. In this architecture, a deep neural network is used to extract a single identity spoofing vector per utterance from the speech features. Then, the extracted vectors are passed to a classifier in order to make the final detection decision. Experimental evaluation is carried out on the standard ASVspoof2015 data corpus. The results show that classical FBANK features and Linear Discriminant Analysis (LDA) obtain the best performance for the proposed system.

Index Terms: Automatic speaker verification, spoofing detection, deep neural networks, deep features, classifier.

1. Introduction

Automatic Speaker Verification (ASV) aims to authenticate the identity claimed by a given individual [1]. However, most ASV systems are vulnerable to spoofing attacks, in which an imposter tries to gain fraudulent access to the system by presenting to the ASV system speech acoustics resembling the voice of a genuine user. Four types of spoofing attacks have been identified [2]: (i) replay (i.e. using pre-recorded voice of the target user), (ii) impersonation (i.e. mimicking the voice of the target voice), and also either (iii) text-to-speech synthesis (TTS) or (iv) voice conversion (VC) systems to generate artificial speech resembling the voice of a legitimate user. The aim of this work is to develop robust anti-spoofing countermeasures for either VC or TTS based attacks.

The performance of anti-spoofing systems can meaningfully vary depending on the voice features used to feed them. Due to this, voice features have attracted the attention of a number of researchers [8, 9, 10]. However, anti-spoofing systems based on neural networks usually use classical voice features, such as FBANKs, and to the best of our knowledge, the new popular CQCC features have not been employed yet to feed these types of systems.

In the last years, the technique of deep features extraction have been explored to obtain more discriminative and effective features for spoofing detection [6, 7, 11]. This technique consists of employing deep neural networks in the front-end of the anti-spoofing system which are fed by speech features, so that the deep features extracted by the neural network are passed to a classifier in order to make the final detection decision (genuine or spoof). The core idea is to take advantage of the non-linear modeling and discriminative capabilities of deep neural networks which have shown to be suitable for feature engineering [3], not only for spoofing detection, but also for speech recognition [4], speaker recognition [3], and speech synthesis [5].

In this work, we compare the performance of different features and back-ends in an anti-spoofing system which extracts deep features [6] in order to detect VC and TTS attacks. This anti-spoofing system employs a convolutional neural network (CNN) plus a recurrent neural network (RNN) and gets a single spoofing identity representation per utterance. Although a similar comparison has already been studied in [7], our study presents three important differences: (1) our anti-spoofing system employs a CNN to extract convolutional features at the speech frame level, (2) we compare the performance of classical features, such as FBANKs and MFCCs, with the performance of the recent popular CQCC features [8], and (3) we combine different features and classifiers in order to find the combination which offers the best performance.

This paper is organized as follows. Section 2 describes the features and back-ends we are going to compare in a CNN + RNN anti-spoofing system. Then, in Section 3, we outline the speech corpora, the network training, and the performance evaluation details. Section 4 discusses the results of the different features and back-ends in the deep neural network based anti-spoofing system. Finally, we present the conclusions derived from this research in Section 5.

2. System description

This section is devoted to the description of the anti-spoofing system. First, Section 2.1 describes different voice features: FBANK, MFCC and CQCC. The neural network architecture for deep feature extraction is detailed in Section 2.2. Furthermore, Section 2.3 describes different classifiers (back-ends): Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), and One-Class Support Vector Machine (One-Class SVM).

2.1. Speech features

As demonstrated in [11], traditional log MEL filterbank features (FBANK) are effective for detecting spoofing attacks with systems based on neural networks. These features are obtained by passing the Short Time Fourier Transform (STFT) magnitude spectrum through a Mel-filterbank and applying a log opera-
However, FBANK features are usually high-correlated. One way to decorrelate these features is to apply the Discrete Cosine Transform (DCT) to get the classical Mel Frequency Cepstral Coefficient (MFCC) features.

In [8], CQCC features are proposed for spoofing detection, which are obtained using the Constant Q Transform (CQT). The Q factor is a measure of the selectivity of each filter and is defined as the ratio between the center frequency and the bandwidth of the filter. In contrast to the STFT, whose Q factor increases when moving from low to high frequencies as the bandwidth is the same for all filters, the bandwidth of the filters employed in the CQT is not constant, and this results in getting a higher frequency resolution for low frequencies and a higher temporal resolution for high frequencies. In this manner, the CQCC features try to imitate the human perception system which is known to approximate a constant Q factor between 500Hz and 20kHz [20].

In this work, we employ the classical FBANK and MFCC features, as well as the popular CQCC features, to feed the anti-spoofing system.

2.2. Front-end

The front-end architecture of the anti-spoofing system is shown in Fig.1. A context window of W frames (centered at the frame being processed) is used to obtain the input signal spectral features which are fed into the system. Then, the CNN provides a deep feature vector per window, and all deep features vectors of the considered utterance are processed by the RNN which computes an embedding vector for the whole utterance. We call this the spoofing identity vector. Since the front-end is trained to perform utterance-level classification of the attacks, this embedding vector should provide more discriminative information for spoofing detection than the raw speech features.

In this architecture, the CNN plays the role of a frame-level deep feature extractor providing one feature vector for each context window of W frames. In order to this, the CNN acts as a classifier whose task consists of determining whether the input feature are either genuine or belong to one of the K spoofing attacks (S1, S2, ..., SK) present in the training set. This CNN uses 2 convolutional and pooling layers as feature extractors, followed by 2 fully connected layers with a softmax layer of K + 1 neurons as classification layer. To prevent overfitting, we used an annealed dropout training procedure [17]. In annealed dropout, the dropout probability of the nodes in the network is decreased as training progresses. In this work, the annealed function reduces the dropout rate from an initial rate prob[0] to zero over N steps with constant rate. The dropout probability prob[t] at epoch t is given as:

\[ prob[t] = \max (0, 1 - \frac{t}{N}) \cdot prob[0]. \]  

As shown in Fig. 1, the deep features obtained from the CNN are fed into an RNN, which computes the anti-spoofing identity vector of the utterance. The main advantage of using an RNN, based on gated recurrent units (GRU) [16], is its ability for learning the long-term dependencies of the subsequent deep feature vectors. Finally, a fully connected layer containing K + 1 neurons (one per class: genuine, S1, S2, ..., SK) is connected to the output of the last time step, followed by a softmax layer. The state of the last time step represents the single identity spoofing vector of the whole utterance.

2.3. Back-end

After deep feature extraction, every utterance is represented by a single spoofing identity vector. A back-end classifier is then applied on these vectors to do the final detection decision. In this section three classifiers will be tested: LDA, SVM and one-class SVM.

A. Linear Discriminant Analysis

LDA assumes that each class density can be modeled as a multivariate Gaussian

\[ N(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}, \]  

where \( \Sigma_k \) and \( \mu_k \) is the covariance and mean for class \( k \), and \( p \) is the dimension of the identity vectors. Moreover, the LDA model assumes every class shares the same covariance, that is, \( \Sigma_k = \Sigma, \forall k \). The goal of LDA is to find a transformation which maximizes the distance between classes while minimizing the spreading within each class. This can be formulated as a diagonalization problem where matrix \( \Sigma_k \Sigma \) (\( \Sigma_k \) is the between-class covariance) is diagonalized, so the transformation can be built from the resulting eigenvectors.

Our LDA classifier uses \( K+1 \) classes which represent genuine speech and the \( K \) known spoofing attacks considered in the training set. In this way, the LDA assigns a genuine speech confidence score to each utterance, which is then used for binary decision (spoof or genuine) during the evaluation.

B. Support Vector Machine

A support vector machine (SVM) separates data points in a high dimensional space defined by a kernel function. In this manner, we first obtain a binary function that describes the probability density function where the genuine data lives. This function returns +1 in the small region corresponding to the genuine speech data and -1 elsewhere. Thus, the core idea of SVM is to estimate the hyperplane with the largest separation margin between the two classes.
In this work, this classifier is used to classify the spoofing identity vectors obtained by the front-end system, where +1 indicates genuine speech and -1 indicates spoofed speech.

C. One-Class Support Vector Machine

Complex classifiers may overfit the training spoofed data. To create a spoof-independent system, we also test a derivative model that can only be trained on genuine speech data. This is a type of one-class SVMs [12], usually employed to find abnormal data. This was first tried in spoofing detection with phase-based features in [13]. This kind of SVM is also applied here to classify the spoofing identity vectors, and only genuine speech data has been used to train the one-class SVM model.

3. Experimental framework

To evaluate the performance of several features and back-ends in an anti-spoofing system based on neural networks, the ASVspoof 2015 dataset [14], a standard data corpus for research on spoofing detection, was employed. Details about the methodology followed for training and testing are also given in this section.

3.1. Speech corpus

The ASVspoof 2015 corpus [14] defines three datasets (training, development and evaluation), each one containing a mix of genuine and spoofed speech. The structure of these three datasets is shown in Table 1. Spoofing attacks were generated either by TTS or VC. A total of 10 types of spoofing attacks (S1 to S10) are defined: three of them are implemented using TTS (S3, S4 and S10), and the remaining seven ones (S1, S2, S5, S6, S7, S8 and S9) using different VC systems. Attacks S1 to S5 are referred to as known attacks, since the training and development sets contain data for these types of attacks, while attacks S6 to S10 are referred to as unknown attacks, because they only appear in the evaluation set. More details about this corpus can be found in [14].

3.2. Spectral Analysis

The frame window size is 25 ms with 10 ms of frame shift. Moreover, the size of the context window is $W = 31$ frames, and the number of filters used to get the spectral features is $M = 48$ filters. In contrast to [7] and [11], we use a 48-dim static spectral features without delta and acceleration coefficients, as we have realized that the context window of 31 frames is already exploiting the correlations between consecutive frames. Therefore, a higher spectral resolution is achieved while the size of the spectral feature vector is smaller than in [7].

3.3. Training

The CNN and RNN networks are trained using Adam optimizer [18]. As there are $K = 5$ known spoofing attacks in the data corpus, the softmax layer of both CNN and RNN contains $K + 1 = 6$ neurons (one per class). The two fully connected layers of the CNN have 1024 sigmoid neurons, and the layer of the RNN has 1920 GRUs, which is the length of the identity spoofing vector of the whole utterance. To prevent the problem of overfitting, the initial dropout probabilities are 50% and 40% from the first to the last fully connected layer, respectively. Also, early stopping is applied in order to stop the training process when no improvement of the cross entropy is obtained after 15 iterations. All the specified parameters of the system have been optimized using the validation set of the data corpus [14].

3.4. Performance evaluation

The equal error rate (EER) is used to evaluate the system performance. As described in the ASVspoof 2015 challenge evaluation plan [14], the EER was computed independently for each spoofing algorithm and then the average EER across all attacks was used. To compute the average EER, we used the Bosaris toolkit [15].

4. Experimental results

4.1. Comparison of features and back-ends

Table 2 shows the detailed results of the different features (FBANK, MFCC and CQCC) and classifiers (LDA, SVM and SVM One-Class) in the described CNN + RNN anti-spoofing system. Furthermore, a summary of these results is shown in Fig. 2. The best performance is obtained with the combination of FBANK features and the LDA classifier. In average, the FBANK features obtain the best performance independently of the back-end, although MFCC features perform better on the SVM One-Class considering all the attacks. The CQCC features achieve the best average performance in the known attacks with LDA and SVM back-ends, but these two combinations perform very poorly in the S10 attack.

Regarding the back-ends, the LDA outperforms the other 2 classifiers in the known and unknown attacks. Moreover, the binary SVM classifier performs much better than SVM One-Class using FBANK and CQCC features.
Table 2: Comparison on evaluation dataset for each spoofing attack in terms of (%) EER

<table>
<thead>
<tr>
<th>Features</th>
<th>Back-end</th>
<th>Known Attacks</th>
<th></th>
<th></th>
<th></th>
<th>Unknown Attacks</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Total</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>S4</td>
<td>S5</td>
<td>Avg.</td>
<td>S6</td>
<td>S7</td>
<td>S8</td>
<td>S9</td>
<td>S10</td>
<td>Avg.</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>S4</td>
</tr>
<tr>
<td>FBANK</td>
<td>LDA</td>
<td>0.01</td>
<td>0.09</td>
<td>0.00</td>
<td>0.00</td>
<td>0.11</td>
<td>0.04</td>
<td>0.66</td>
<td>0.21</td>
<td>0.00</td>
<td>0.36</td>
<td>7.16</td>
<td>1.68</td>
<td>0.86</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.03</td>
<td>0.13</td>
<td>0.00</td>
<td>0.01</td>
<td>0.22</td>
<td>0.08</td>
<td>0.77</td>
<td>0.34</td>
<td>0.18</td>
<td>0.48</td>
<td>10.46</td>
<td>2.44</td>
<td>1.26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVMOne</td>
<td>0.36</td>
<td>2.07</td>
<td>0.17</td>
<td>0.12</td>
<td>4.37</td>
<td>1.42</td>
<td>5.44</td>
<td>1.34</td>
<td>0.34</td>
<td>1.53</td>
<td>8.23</td>
<td>3.38</td>
<td>2.40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MFCC</td>
<td>LDA</td>
<td>0.06</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
<td>0.06</td>
<td>0.04</td>
<td>0.11</td>
<td>0.12</td>
<td>0.00</td>
<td>0.05</td>
<td>15.43</td>
<td>3.14</td>
<td>1.59</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.05</td>
<td>0.19</td>
<td>0.01</td>
<td>0.01</td>
<td>0.23</td>
<td>0.10</td>
<td>0.22</td>
<td>0.21</td>
<td>0.05</td>
<td>0.15</td>
<td>29.58</td>
<td>6.04</td>
<td>3.07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVMOne</td>
<td>0.43</td>
<td>1.97</td>
<td>0.12</td>
<td>0.12</td>
<td>2.11</td>
<td>0.95</td>
<td>3.38</td>
<td>2.07</td>
<td>0.06</td>
<td>1.03</td>
<td>11.09</td>
<td>3.53</td>
<td>2.24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CQCC</td>
<td>LDA</td>
<td>0.04</td>
<td>0.04</td>
<td>0.00</td>
<td>0.00</td>
<td>0.04</td>
<td>0.02</td>
<td>0.13</td>
<td>0.51</td>
<td>0.05</td>
<td>0.08</td>
<td>11.76</td>
<td>2.51</td>
<td>1.27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.03</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
<td>0.01</td>
<td>0.06</td>
<td>0.37</td>
<td>0.07</td>
<td>0.02</td>
<td>21.52</td>
<td>4.41</td>
<td>2.21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVMOne</td>
<td>1.72</td>
<td>6.14</td>
<td>0.49</td>
<td>0.47</td>
<td>7.34</td>
<td>3.23</td>
<td>10.13</td>
<td>9.67</td>
<td>1.39</td>
<td>6.50</td>
<td>8.54</td>
<td>7.25</td>
<td>5.24</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Comparison on evaluation dataset for known and unknown spoofing attacks in terms of average (%) EER

According to these results, we propose an anti-spoofing system which employs FBANK features, a CNN + RNN architecture to get the spoofing identity vector of an utterance, and an LDA classifier to make the final detection decision (spoof or genuine).

4.2. Comparative performance

A comparison of our proposal with other popular techniques from the literature are presented in Fig. 3.

The first two systems CQCC + GMM [8] and CFCC-IF + GMM [9] employ the features which perform best for spoofing detection using a Gaussian Mixture Model (GMM) as back-end. The other four systems are the most popular anti-spoofing systems based on deep learning frameworks. The FBANK + CNN + LDA system has been proposed in [11], but as its performance is not provided in this reference for the clean scenario, we have evaluated instead our proposed system removing the RNN and averaging the deep features for getting the spoofing identity vector of the utterance as in [11].

The CQCC + GMM system achieves the best average performance, although our proposed system (FBANK + CNN + RNN + LDA) achieves the best results for the known attacks. Compared to the rest of deep learning systems (Spectro + CNN + RNN [19], FBANK + DNN + LDA [7], FBANK + RNN + SVM [7] and FBANK + CNN + LDA), our proposal outperforms all of them for the known and unknown attacks. In particular, the result of our proposal for the S10 attack is quite noteworthy. Furthermore, our proposed system also achieves a lower EER in almost all attacks than the CFCC-IF system [9], performing 0.45% better on average when considering all the attacks.

Despite that the CQCC + GMM system outperforms all the systems in a clean condition training scenario, our previous work [6] and reference [11] demonstrate that CQCC + GMM performs worse than the systems based on deep features when a noisy scenario is considered. Moreover, reference [6] shows that, when a typical multicondition training for noise robustness is applied, our system, based on deep identity vectors, clearly outperforms CQCC + GMM even in clean conditions.

5. Conclusions

This paper has evaluated different features and classifiers in order to find the combination which offers the best performance for an anti-spoofing system based on a deep learning framework. The experimental results have shown that FBANK features and an LDA obtain the best performance for systems based on the extraction of deep features, rather than the popular CQCC features and other types of classifiers, such as binary SVM and SVM One-Class. Furthermore, the proposed system (FBANK + CNN + RNN + LDA) outperforms the rest of deep learning systems of the literature.

6. Acknowledgements

This work has been supported by the Spanish MINECO/FEDER Project TEC2016-80141-P and the Spanish Ministry of Education through the National Program FPU (grant reference FPU16/05490). We also acknowledge the support of NVIDIA Corporation with the donation of a Titan X GPU.
7. References


The observation likelihood of silence: analysis and prospects for VAD applications

Igor Odriozola¹, Inma Hernaez¹, Eva Navas¹, Luis Serrano¹, Jon Sanchez¹
¹AHLAB, University of the Basque Country UPV/EHU
{igor, inma, eva, lserrano, jon}@aholab.ehu.eus

Abstract
This paper shows a research on the behaviour of the observation likelihoods generated by the central state of a silence HMM (Hidden Markov Model) trained for Automatic Speech Recognition (ASR) using cepstral mean and variance normalization (CMVN). We have seen that observation likelihood shows a stable behaviour under different recording conditions, and this characteristic can be used to discriminate between speech and silence frames. We present several experiments which prove that the mere use of a decision threshold produces robust results for very different recording channels and noise conditions. The results have also been compared with those obtained by two standard VAD systems, showing promising prospects. All in all, observation likelihood scores could be useful as the basis for the development of future VAD systems, with further research and analysis to refine the results.

Index Terms: VAD, observation likelihood, cepstral normalization

1. Introduction
Voice activity detection (VAD) is an important issue in Automatic Speech Recognition (ASR) or ASR-based systems. It allows the systems to reduce the computation cost and, as a consequence, the response time of the decoding process, by only passing speech frames [1]. If the access to the system is intended to be universal, the VAD has to cope with different noise levels, with no—or little—loss in accuracy. Indeed, the greatest challenge for the current ASR systems is to cope with background noise in the input speech signal [2].

A large number of speech features and combinations have been proposed for VAD [3]. Gaussian Mixture Models (GMM) and Hidden Markov Models (HMMs) have been tested in this context [4][5]. Recently, the use of classifiers has been very common: decision trees (DT) [6], Support Vector Machines (SVM) [7] and hybrid SVM/HMM architectures [8]. More recently, neural networks (NN) have appeared in the literature outperforming the previous designs [9][10][11]. However, these approaches are complex and do not work in real time.

Little research has been done using cepstral normalization for VAD proposals, although it proved to be rather discriminative already in [12]. Here, we introduce some research on the use of observation likelihoods for VAD, applying Cepstral Mean and Variance Normalization (CMVN). We analyse the behaviour of the observation likelihoods generated by the GMM in the central state of the silence HMMs trained for ASR. Results show that it is a promising basis for future prospects.

The next section is a study of different aspects of the observation likelihood scores. Section 3 describes the databases and metrics used for the experiments. Then, VAD some experiments are shown in section 4. Finally, some conclusions and future prospects are explained in section 5.

2. The observation likelihood
In speech recognition, audio segments corresponding to the same recognition unit (word, phone, triphone etc., even silence or non-speech) are gathered and processed, in order to extract acoustic parameters from them —typically Mel-frequency cepstral coefficients (MFCC)— and train a different acoustic model for each unit. A very popular acoustic model is the HMM, since it not only models the likelihood of a new observation vector, but also the sequentiality of the observations.

Usually, observation likelihoods are generated by the GMM belonging to each HMM state $j$. For an observation vector $\mathbf{y}_n$, the observation likelihood $b_j$ of a GMM is calculated as shown in equation 1.

$$b_j (\mathbf{y}_n) = \sum_{m=1}^{M} c_{jm} N (\mathbf{y}_n; \mu_{jm}, \Sigma_{jm})$$

(1)

where $M$ is the number of mixture components, $c_{jm}$ is the weight of the $m$th component and $N (\cdot; \mu, \Sigma)$ is a multivariate Gaussian with mean vector $\mu$ and covariance matrix $\Sigma$.

In this work, the observation likelihoods have been obtained from the silence HMM trained using the Basque Speechon-like database [13], specifically the close-talk channel.

2.1. The acoustic model for silence
The HMM topology chosen for silence frames has three states, left-to-right, allowing the right-end state to connect back with the left-end state. It was trained with 13 MFCCs and 13 first and second order derivatives as acoustic parameters, and 32-mixture GMMs. The frame length is 25 ms with a shift of 10 ms.

CMVN was applied to MFCCs, computing global means and variances from each recording session. For $N$ cepstral vectors $\mathbf{y} = \{ y_1, y_2, ..., y_N \}$, their mean $\mu_N$ and variance $\sigma_N^2$ vectors are calculated as defined in equations 2 and 3, respectively.

$$\mu_N (i) = \frac{1}{N} \sum_{n=1}^{N} y_n (i)$$

(2)

$$\sigma_N^2 (i) = \frac{1}{N} \sum_{n=1}^{N} (y_n (i) - \mu_N (i))^2$$

(3)

where $i$ is the $i$th component of the vector.

The cepstral features are then normalized using the calculated mean and variance vectors, as given in equation 4. Thus, each normalized feature has zero mean and unit variance.

$$\hat{y}_n (i) = \frac{y_n (i) - \mu_N (i)}{\sigma_N (i)}$$

(4)
2.2. The impact of CMVN

The use of CMVN has a significant impact on the curves that observation likelihoods form. When testing a sample signal and computing frame by frame the observation likelihoods at each state of the silence HMM, very different curves are obtained depending on whether CMVN is applied or not. Figure 1 illustrates this difference. The middle and bottom diagrams show the curves formed by the observation log-likelihoods generated by each HMM state $s_0$, $s_1$, and $s_2$, without and with normalization respectively, through a utterance composed of four words. In this case, the normalization has been performed using the means and variances computed from the file.

![Figure 1: Spectrogram (top) and observation log-likelihoods along time (frames) generated by the left state ($s_0$), central state ($s_1$) and right state ($s_2$) of the silence HMM without CMVN (middle) and applying CMVN (bottom).](image)

The curves in the bottom diagram (with CMVN), compared with the ones in the middle diagram (without CMVN), look more abrupt. This fact can be used to better discern between speech and non-speech.

2.3. The central state of the silence HMM

In any three-state HMM, the central state is a priori the most stable state of the model, since the left and right states have to cope with transitions between models. It makes sense that the same will happen to the silence HMM, where left and right states have to model transitions between silence and speech.

Looking back at Figure 1, we can see that, indeed, the curves generated by the central state ($s_1$) are, in both cases (with and without cepstral normalization), much more discriminative than the curves corresponding to the states at the ends, which are more irregular.

2.4. Robustness against different $SNR$ values

Another interest point to focus on in a VAD is its robustness for different recording conditions. As an example, we have chosen four signals from the Spanish Speecon database [14] to illustrate the impact of the recording distance on the observation likelihood curves. These four signals correspond to the same utterance, but were recorded by means of four different microphones: a headset (channel $C_0$), a lavalier (channel $C_1$), a medium-distance cardioid microphone (0.5-1 meter, channel $C_2$) and a far-distance omnidirectional microphone (channel $C_3$). Each of these channels represents a different $SNR$, $C_0$ being the cleanest (around 20dB) and $C_3$ the noisiest (0dB).

Figure 2 shows the observation log-likelihoods generated by the central state of the silence HMM trained with the Basque Speecon-like database. The utterance is the same as the one in Figure 1 (note that the signal in Figure 1 corresponds to the $C_1$ signal in Figure 2). The darkest curve corresponds to the $C_0$ channel and the lightest one to the $C_3$ channel.

![Figure 2: Observation log-likelihoods along time obtained at the central state ($s_1$) of the silence HMM when processing different channels ($C_0, C_1, C_2, C_3$).](image)

The curves show that, as expected, a degradation occurs when the signals recorded at farther distances are processed, but even so the curves remain rather discriminative. For $C_3$ signals, the most adverse effect occurs at the initial and ending phones, where, depending on the phone, likelihoods can be very similar to those of the noisy silence. This happens mostly when the initial phone is a noisy phone. However, the curves show a good behaviour for $C_1$ and $C_2$, with likelihood profiles very similar to those obtained for $C_0$ signals.

3. Data preparation

To assess the stability of the observation likelihood curves generated by the central state of the silence HMM, a VAD accuracy experiment has been carried out, setting different thresholds to label frames as speech or silence.

3.1. The databases

Two databases have been chosen for the experiments: first, the Noisy TIMIT speech database [15], to analyse weather a threshold could be set for different $SNR$ conditions. The second database is the ECNESS subset of the Spanish Speecon database [16], which has been used to test the validity of that threshold.

1. Noisy TIMIT speech database: it contains approximately 322 hours of speech from the TIMIT database [17] modified with different additive noise levels. However, we have chosen only babble and white noises, as the most natural ones. Noise levels vary in 5 dB steps and ranging from 50 to 5 dB. The database contains 630 different
speakers, with 10 utterances per speaker: 6300 files for each noise level. The total speech content in the database is 86.57% (not well balanced), and the label files are the ones belonging to the classic TIMIT database. All audio files are presented as single channel 16kHz 16-flac, but have been converted to 16-bit PCM.

2. ECESS subset of the Spanish Speeccon database: it was used in the ECESS evaluation campaign of voice activity and voicing detection in 2008. It includes 1020 utterances recorded in different environments (office, entertainment, car and public place) distributed among the $C_0$, $C_1$, $C_2$ and $C_3$ subsets (total number of files: 4080). There are 60 different speakers each of which utters 17 sentences. The total speech content in the database is 55.77% (well balanced), and it contains reference speech and silence labels specifically designed to assess different VAD algorithms. The signals in the database were recorded at 16 kHz and 16 bit per sample.

Each file’s features have been normalized off-line, with the means and variances calculated from the file itself. The on-line performance has been left for future research.

3.2. Error metrics
The VAD accuracy experiment consists in evaluating the ability of the system to discriminate between speech and silence segments at different SNR levels, in terms of silence error-rate ($ER_0$) and speech error-rate ($ER_1$). These two rates are computed as the fractions of the silence frames and speech frames that are incorrectly classified ($N_{0,1}$ and $N_{1,0}$, respectively) among the number of real silence frames and speech frames in the whole database ($N_{0}^{s/f}$ and $N_{1}^{s/f}$, respectively), as shown in equation 5. In addition, the $TER$ (total error rate) has also been computed as the average of the $ER_0$ and $ER_1$ (equation 6).

$$ER_0 = \frac{N_{0,1}}{N_{0}^{s/f}} \times 100; \quad ER_1 = \frac{N_{1,0}}{N_{1}^{s/f}} \times 100$$  \hspace{1cm} (5)

$$TER = \frac{ER_0 + ER_1}{2}$$  \hspace{1cm} (6)

A minimum duration of 15 frames both for speech and silence segments was set. This value was empirically chosen after some preliminary experiments.

4. VAD experiments
Initially, we have analysed whether a threshold can be set for VAD purposes, considering the various SNR values. Then, we have tested that threshold in a separate database, and, in addition, a validity test has been carried out comparing the results with those obtained with three standard VAD algorithms.

4.1. Analysis of the decision threshold
Different thresholds have been considered to label frames as speech or silence. Results are shown in Figure 3, both for babble noise (left) and white noise (right).

For the cleanest signals ($SNR = 50dB$), the equal error rate ($EER$) points of $ER_0$ and $ER_1$ curves are located near $-200$. However, as the $SNR$ gets lower, the $EER$ points move towards higher values. In the case of white noise, this shift reaches the $-120$ value for $5 dB$.

Figure 3: $ER_0$ and $ER_1$ (top) and $TER$ (bottom) for different decision threshold values when testing the signals of $SNR$ 50 to 5 dB in the babble noise subset (left) and the white noise subset (right) of the Noisy TIMIT database.

Regarding the error rates, the minimum $TER$s are obtained at $Th = -150$, except for 5, 10 and 15 dB in white noise subset, which occur at $-100$. Thus, we can consider the point of $Th = -150$ as the most valid threshold. Some $ER_0$ and $ER_1$ values obtained for $Th = -150$ are shown in Table 1.

Table 1: $TER$, $ER_0$ and $ER_1$ for $Th = -150$ on the signals of $SNR$ 50, 35, 20 and 5 dB in the babble noise (left) and white noise (right) subsets of the Noisy TIMIT database.

<table>
<thead>
<tr>
<th>SNR</th>
<th>$ER_0$</th>
<th>$ER_1$</th>
<th>$TER$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50dB</td>
<td>34.89</td>
<td>6.71</td>
<td>20.80</td>
</tr>
<tr>
<td>35dB</td>
<td>30.87</td>
<td>7.48</td>
<td>19.18</td>
</tr>
<tr>
<td>20dB</td>
<td>26.35</td>
<td>11.25</td>
<td>21.53</td>
</tr>
<tr>
<td>5dB</td>
<td>22.60</td>
<td>20.78</td>
<td>21.70</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNR</th>
<th>$ER_0$</th>
<th>$ER_1$</th>
<th>$TER$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50dB</td>
<td>6.95</td>
<td>34.88</td>
<td>20.92</td>
</tr>
<tr>
<td>35dB</td>
<td>9.18</td>
<td>28.06</td>
<td>18.62</td>
</tr>
<tr>
<td>20dB</td>
<td>16.89</td>
<td>21.53</td>
<td>19.21</td>
</tr>
<tr>
<td>5dB</td>
<td>30.90</td>
<td>15.49</td>
<td>23.20</td>
</tr>
</tbody>
</table>

For $Th = -150$, the minimum $ER_1$ is 6.71, at 50 dB. As expected, the $ER_1$ increases as the $SNR$ decreases. However, notice that the $TER$ does not present the minimum at 50 dB, neither in the babble noise subset nor in the white noise subset, as might be expected.

4.2. Testing
The threshold calculated in the previous section has been applied to the files of ECESS subset of the Spanish Speeccon database. 4080 files have been tested (1020 in each $C_i$ subset). Results are shown in Table 2.

The results obtained for the ECESS subset using the threshold calculated from the Noisy TIMIT are very good. Compared with the best result obtained for the Noisy TIMIT (see 50 dB row in Table 1), much lower $ER_0$ and $ER_1$ have been obtained. The error rates, as expected, increase as $SNR$ decreases, al-
Table 2: TER, ER₀ and ER₁ with \( Th = -150 \) on the signals of channels \( C₀, C₁, C₂ \) and \( C₃ \) in the Spanish Speecon database.

<table>
<thead>
<tr>
<th></th>
<th>( ER₀ )</th>
<th>( ER₁ )</th>
<th>TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C₀ )</td>
<td>6.21</td>
<td>2.74</td>
<td>4.48</td>
</tr>
<tr>
<td>( C₁ )</td>
<td>4.22</td>
<td>6.13</td>
<td>5.18</td>
</tr>
<tr>
<td>( C₂ )</td>
<td>7.10</td>
<td>6.00</td>
<td>6.55</td>
</tr>
<tr>
<td>( C₃ )</td>
<td>9.46</td>
<td>6.45</td>
<td>7.96</td>
</tr>
</tbody>
</table>

though the best silence error rate is obtained for the \( C₁ \) channel. Additionally, a tuning has been performed for \( ER₁ \) reduction. Indeed, for speech processing, it is important to reduce the \( ER₁ \) as much as possible, so that the minimum number of speech frames are lost for the next stage. For that purpose, we have sought to reduce the impact of non-speech to speech boundaries, setting an additional margin of 5 and 10 frames around the speech segments. Results are shown in Table 3.

Table 3: TER, ER₀ and ER₁ for 5 and 10 frames long speech-segment margins, with \( Th = -150 \) for the signals of channels \( C₀, C₁, C₂ \) and \( C₃ \) in the ECESS subset of the Spanish Speecon database.

<table>
<thead>
<tr>
<th></th>
<th>( ER₀ )</th>
<th>( ER₁ )</th>
<th>TER</th>
<th>( ER₀ )</th>
<th>( ER₁ )</th>
<th>TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C₀ )</td>
<td>10.84</td>
<td>1.29</td>
<td>6.07</td>
<td>15.68</td>
<td>0.79</td>
<td>8.24</td>
</tr>
<tr>
<td>( C₁ )</td>
<td>7.94</td>
<td>3.47</td>
<td>5.71</td>
<td>12.42</td>
<td>2.30</td>
<td>7.36</td>
</tr>
<tr>
<td>( C₂ )</td>
<td>10.91</td>
<td>3.50</td>
<td>7.21</td>
<td>15.39</td>
<td>2.47</td>
<td>8.93</td>
</tr>
<tr>
<td>( C₃ )</td>
<td>13.29</td>
<td>3.95</td>
<td>8.62</td>
<td>17.59</td>
<td>2.89</td>
<td>10.24</td>
</tr>
</tbody>
</table>

The table shows that \( ER₁ \) reduces and \( ER₀ \) increases. TER increases as well, because \( ER₀ \) increases faster than \( ER₁ \) reduces. All in all, the use of a margin around speech segments allows decreasing significantly \( ER₁ \), with a not very significant resulting TER degradation.

4.3. Comparison with other systems

In order to validate the previous results, our results have been compared with the outcomes of three popular standard VAD algorithms carried out in a previous work [18]. These systems are standard defined by ITU (International Telecommunication Union) and ETSI (European Telecommunications Standards Institute):

1. The VAD algorithm of the ITU G.729 system [19].
2. The AFE-FD (frame-dropping mechanism) algorithm implemented in ETSI AFE-DSR (Advanced Front-End for Distributed Speech Recognition) [20].
3. The AFE-NR (noise reduction system) algorithm implemented in ETSI AFE-DSR [20].

Table 4 shows the results obtained for the three VAD systems along with the proposed method (using \( Th = -150 \) and a margin of 10 frames), over the same dataset (4080 files from the ECESS subset). Regarding \( ER₁ \), the AFE-FD gets better results, and also the AFE-NR for \( C₀ \) and \( C₁ \). However both systems show the disadvantage of getting very high \( ER₀ \) for all the channels (the lowest value is 38.10 %). This means that many silence frames will be sent to the recognizer. The \( ER₀ \) in our results are between 12.42 and 17.59 %.

Table 4: Comparison of different VAD algorithm results at four SNR levels

(a) Silence error rates \( (ER₀) \)

<table>
<thead>
<tr>
<th></th>
<th>G.729</th>
<th>AFE-FD</th>
<th>AFE-NR</th>
<th>Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C₀ )</td>
<td>56.06</td>
<td>63.88</td>
<td>58.23</td>
<td>15.68</td>
</tr>
<tr>
<td>( C₁ )</td>
<td>70.23</td>
<td>54.75</td>
<td>55.96</td>
<td>12.42</td>
</tr>
<tr>
<td>( C₂ )</td>
<td>59.54</td>
<td>52.10</td>
<td>38.10</td>
<td>15.39</td>
</tr>
<tr>
<td>( C₃ )</td>
<td>70.49</td>
<td>50.10</td>
<td>47.65</td>
<td>17.59</td>
</tr>
</tbody>
</table>

(b) Speech error rates \( (ER₁) \)

<table>
<thead>
<tr>
<th></th>
<th>G.729</th>
<th>AFE-FD</th>
<th>AFE-NR</th>
<th>Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C₀ )</td>
<td>3.63</td>
<td>0.03</td>
<td>0.62</td>
<td>0.79</td>
</tr>
<tr>
<td>( C₁ )</td>
<td>9.28</td>
<td>0.23</td>
<td>1.98</td>
<td>2.30</td>
</tr>
<tr>
<td>( C₂ )</td>
<td>18.19</td>
<td>0.48</td>
<td>4.83</td>
<td>2.47</td>
</tr>
<tr>
<td>( C₃ )</td>
<td>17.22</td>
<td>1.41</td>
<td>8.30</td>
<td>2.89</td>
</tr>
</tbody>
</table>

5. Conclusions

In this paper, we have assessed the usefulness of the observation likelihood generated by the central state GMM of a silence HMM trained using CMVN, as a possible basis on which to build a VAD system. We have seen that a good classification between speech and silence can be performed, just by setting a threshold in the curves that observation likelihoods form.

The silence HMM has been trained using the close-talk channel from the Basque Speecon-like database. Then, a threshold analysis has been carried out, processing the babble and white noise files of the Noisy TIMIT database. As a conclusion, we have noticed that the minimum error rates occur at the same likelihood point in 17 SNR values out of a total of 20. This point is the one we have chosen as the threshold.

This threshold has been tested with a separate database: the ECESS subset of the Spanish Speecon database. The results obtained for this database are even better than those obtained for the Noisy TIMIT, which leads us to think that the silence observation likelihood behaves similarly on different channels.

Additionally, the results of the test have been compared with three different standard VAD systems. Although the best speech error rates have not been achieved with the use of the decision threshold, we have got the best silence error rates. Our results are quite competitive; actually, the best total classification rates have been obtained.

As a final conclusion, competitive results are obtained just by setting a decision threshold to the silence observation likelihood curves. This fact has been applied in [21], where a method called Multi-Normalization Scoring (MNS) is used to explore the discriminative potential of the observation likelihood scores. Robust on-line results are shown in that paper, where the scores obtained with MNS are classified with a Multi-Layer Perceptron (MLP). This issue and others related to the selection of the optimal threshold are being investigated currently in our laboratory.

6. Acknowledgements

This work has been partially supported by the EU (FEDER) under grant TEC2015-67163-C2-I-R (RESTORE) (MINECO/FEDER, UE) and by the Basque Government under grant KK-2017/00043 (BerbaOla).
7. References


On the use of Phone-based Embeddings for Language Recognition

Christian Salamea\textsuperscript{1,2}, Ricardo de Córdoba\textsuperscript{2}, Luis Fernando D’Haro\textsuperscript{2}, Rubén San Segundo\textsuperscript{2}, Javier Ferreiros\textsuperscript{2}

\textsuperscript{1}Interaction, Robotics and Automation Research Group, Universidad Politecnica Salesiana, Calle Vieja 12-30 y Elia Liut, Cuenca, Ecuador.
\textsuperscript{2}Speech Technology Group, Information and Telecommunications Center, Universidad Politecnica de Madrid, Ciudad Universitaria Av. Complutense, 30, 28040, Madrid

csalamea@ups.edu.ec, cordoba@die.upm.es, lfdharo@die.upm.es, lapi@die.upm.es, jfl@die.upm.es

Abstract

Language Identification (LID) can be defined as the process of automatically identifying the language of a given spoken utterance. We have focused in a phonotactic approach in which the system input is the phoneme sequence generated by a speech recognizer (ASR), but instead of phonemes, we have used phonetic units that contain context information, the so-called “phone-gram sequences”. In this context, we propose the use of Neural Embeddings (NEs) as features for those phone-grams sequences, which are used as entries in a classical i-Vector framework to train a multi class logistic classifier. These NEs incorporate information from the neighbouring phone-grams in the sequence and model implicitly longer-context information. The NEs have been trained using both a Skip-Gram and a Glove Model. Experiments have been carried out on the KALAKA-3 database and we have used Cavg as metric to compare the systems. We propose as baseline the Cavg obtained using the NEs as features in the LID task, 24.7%. Our strategy to incorporate information from the neighbouring phone-grams to define the final sequences contributes to obtain up to 24.3% relative improvement over the baseline using Skip-Gram model and up to 32.4% using Glove model. Finally, the fusion of our best system with a MFCC-based acoustic i-Vector system provides up to 34.1% improvement over the acoustic system alone.

Index Terms: language identification, phonotactic, neural embeddings

1. Introduction

Automatic spoken language identification (LID) is the process of identifying the actual language of a sample of speech using a known set of trained language models [1]. There are currently two main ways of achieving this goal: the first one uses acoustic features extracted from the speech signal in which the spectral information is used to distinguish between languages, while the second method uses the phonetic sequences obtained using an automatic phonetic recognizer (ASR) as features.

In general, the best results for the LID task are achieved using acoustic-based systems. However their fusion with phonetic/phonotactic-based systems provides a higher accuracy [2]. This paper focuses on the study of phonotactic techniques, but we will also show that it contributes to an improvement in the overall system thanks to the fusion [3] of both techniques.

Neural Embeddings (NEs) are vector representations [4] of the phonetic units and have been used in speech recognition tasks [5]. These vector representations are extracted from either the hidden layer in an Neural Network (NN) [6] or from the occurrence matrix [7] of phonetic units in an unlabeled corpus. NEs has been successfully used in speech recognition systems at a word level [5], because of their ability to model the probability of one word appearing in a context close to another one, however their suitability in phonotactic LID systems has not yet been considered, probably due to some difficulties at the phoneme level that do not appear at the word level.

For example, NEs are normally created to reflect the relation at the semantic and syntactic level between words [8], characteristics that do not exist at a phonetic level. Our proposal is to process NEs similarly to the i-Vector framework, as continuous vectors that contain the most representative information about a language in a low dimension [9]. Our expectation is that NEs in a language will be projected into some particular direction being this projection discriminative in comparison to the other languages.

To overcome the limitations in NEs at the phoneme level, we propose the use of phonetic units that incorporate context information (phone-grams). However, both the high-order phone-grams and the embedding vectors size could make it difficult the training of the i-Vectors because of the large size of the matrix needed for this implementation. We will show two alternatives for dealing with the feature vectors.

Context information has also been incorporated in those feature vectors which we have called “Context Neural Embedding Sequences”, which we have used to generate the i-Vectors. These i-Vectors are used as features of a multiclass logistic classifier that obtain the detection cost values (Cavg) for the language trained.

Finally, all of the systems are fused obtaining the final global Cavg for all of the languages trained.

This paper is organized as follows. In section 2, we describe the different techniques, as well as the acoustic system used in the fusion with the phonotactic system. In section 3, we present the experiments and the final results. Finally, in section 4, the conclusions and future work are presented.
2. System Description

2.1. Phone-gram definition

Phonotactic systems use context information to improve the performance of LID. In this regard, we propose to use phonetic units that implicitly incorporate context information as features (phone-grams). They can be defined as the grouping of two or more phonemes in a new unit (Figure 1). In this work we have used 2grams only because of the scattering observed in higher order.

![Figure 1: Concept of phone-gram.](image)

2.2. Database

The KALAKA-3 database was created for the Albayzin 2012 LRE [10]. It is designed to recognize up to 6 languages in the closed set condition (i.e. Basque, Catalan, English, Galician, Portuguese, and Spanish) using noisy and clean files with an average duration of 120s and includes 108 hours in total. This database contains training, development, testing, and evaluation examples distributed as shown in Table 1:

<table>
<thead>
<tr>
<th>Nº Files</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>4656</td>
<td>458</td>
<td>459</td>
<td>941</td>
<td></td>
</tr>
<tr>
<td>Nº of clean files</td>
<td>3060</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Nº of noisy files</td>
<td>1596</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Length&lt;=30s</td>
<td>2855</td>
<td>121</td>
<td>113</td>
<td>267</td>
</tr>
<tr>
<td>Length 120s</td>
<td>1801</td>
<td>337</td>
<td>346</td>
<td>674</td>
</tr>
</tbody>
</table>

To compare the systems, we have considered the Cavg metric, that weights the number of the false acceptances and false alarms generated by the recognizer, representing them as a detection cost function [11]. Using this metric, lower values correspond to better systems.

2.3. Phoneme Recognizers

The phoneme sequences of the utterances used to obtain the “phone-grams” have been generated from a phoneme recognizer. In our system, it is based on the system designed by the Brno University (BUT) [12], which uses monophone three state HMMs. There are 3 sets of HMMs (for Hungarian, Russian, and Czech) with 61, 46, and 52 different phonemes respectively.

2.4. Phonetic Vector Representation

In our approach, we have called SG-Emb to the vector representations extracted from the hidden layer of an NN and GI-Emb to the vector representations extracted from the co-occurrence of elements matrix obtained from the global corpus. In the first case, the objective is to predict the phonetic unit that is going to appear next according to the context in which the unit is included. In the second case, the objective is to normalize the counts and then smooth them trying to obtain a vector representation with homogeneous values.

The model definition normally used to train both is focused at the word-level [13] but we work at the phone level. The objective is to find the co-occurrence of phonemes and phoneme sequences that tend to appear in similar contexts for a specific language. Hence, we expect to improve the results compared with the system based on uniphone sequences. Our study focuses on phone-grams, and their use in the continuous space has been called Phone-based Embeddings (Ph-Emb).

2.5. Skip-Gram Model

In relation to the SK-Emb obtained from a NN, they are obtained with the following procedure: we consider a NN with one input layer, one hidden layer and one output layer. In the input layer we have the phone-gram with 1-of-N coding, in the state layer we obtain the vector representation of the phone-gram, and the SK-Emb are generated by applying a modelling technique on the vector representation of the input phone-gram together with its context. These SK-Emb will be the feature vectors that we will use in our LID system. From several models that have been proposed for that purpose, two of them are the most used: Skip-Gram [14] and C-Bow [15]. We have selected Skip-Gram as we obtained better results in initial experiments.

The Skip-Gram model is a classic NN, where the activation functions are removed and hierarchical Soft-max [16] is used instead of soft-max normalization. The training objective of the Skip-Gram model is to predict the context of the input phone-gram in the same sentence [17].

2.6. GloVe Model

On the other hand, the GI-Emb from the co-occurrence matrix have been obtained as follows: the matrix contains the counts of a phonetic unit appearing close to a possible context. Rows are the phonetic units of the vocabulary while columns are the possible context of those phonetic units. The least squares model is used in the training process (GloVe model) [7].

GloVe models capture the statistics of the global corpus to learn the vector representations of words. It is normally used at a word level, but in our case, we are going to evaluate them at a phonetic level using phone-grams. GloVe models are similar to Skip-Gram models except for the context windows. GloVe uses the co-occurrence of elements, capturing the global statistics of the corpus in a matrix and analyzing the entire possible co-occurrence probability rates using test phonetic units. This way, it is possible to distinguish the relevant and non-relevant phonemes in a sentence [7].

2.7. i-Vector system based on Phone-based Embeddings applied to LID

We propose to use these Phone-based Embeddings as feature vectors for the input of a LID system based on the i-Vector framework [18]. In Figure 2, we present the global system architecture. Our system has two components. The first one called “Front-End”, is where the acoustic signal processing is carried out followed by the phoneme recognizer,
that produces the phonetic sequences corresponding to the utterances. 

The second component of the system is the "Back-End". Firstly, we obtain the phone-gram sequences from the phoneme sequences of each language. The sequences obtained have been used to train the Phone-based Embeddings. To model the Phone-based Embeddings we have used both alternatives described above, Skip-Gram and GloVe modeling. After that, we have replaced every phone-gram by its respective Phone-based Embedding to use it as input feature vector to the i-Vector system. All these vectors are used to train the T matrix and the UBM model needed to obtain the i-Vectors. Finally we have used these i-Vectors as features to train a multiclass logistic classifier to define the detected language [19], [20].

As we have one different Phone-based Embedding for each language to be recognized, we considered two alternatives to manage the set of vectors for each phone-gram. We have called them: "Single vector embedding" and "Multiple vector embeddings".

2.8. Single Vector Embedding (SVE)

We organize the phone-gram sequence in a column, replacing each phone-gram by its corresponding Phone-based Embedding of a specific language and repeat this process for all the other languages. So, the first column will contain the Phone-based Embeddings sequences trained with data from language 1, the second one will contain the Phone-based Embeddings sequences trained with data from language 2, and so on. Finally, we obtain a matrix that includes the Phone-based Embeddings trained with all the languages to be recognized (Figure 3).

![Figure 3: Single Vector Embedding.](image)

2.9. Multiple Vector Embedding (MVE)

As SVE could easily have a problem of excessive dimensionality, we considered a system where we use the Phone-based Embeddings trained for each language individually (Language Phone-based Embeddings) to obtain different i-Vectors for each language. Our proposal is to fuse the scores provided by the individual language-dependent systems expecting a better performance and a lower computational cost.

2.10. Acoustic System using MFCCs

We have fused the scores of the proposed techniques with the scores obtained from an acoustic system to check if they provide complementary information. The acoustic system has been generated as follows: from each speech utterance, 12 MFCC coefficients including C0 [21] are extracted for each frame. The silence and noise segments of the acoustic signal have been removed using a Voice Activity Detector. To reduce the noise perturbation, a RASTA filter has been used together with a cepstral mean and variance normalization (CMNV). We have a feature vector of dimension 56, generated from the concatenation of the SDC parameters using the 7-1-3-7 configuration. Feature vectors are used to train the total variability matrix, from which the i-vectors of dimension 400 with 512 Gaussians are extracted (optimal configuration).

3. Results

We have to define the Phone-based Embeddings optimal training parameters for 2grams: vector size, window size, number of training iterations and negative sampling factor. Negative sampling is an optimization method used to improve the NEs robustness applying logistic regression. It reduces the computational complexity and increases the vector estimated efficiency.

The window size corresponds to the number of phonetic phone-gram units considered to the left and to the right of the current phonetic unit and it is considered as contextual information. The vector size is the vector embedding size. In all cases, the results in the tables represent the fusion of the three phonetic recognizers.

3.1. Single Vector Embedding (SVE)

As we described in Section 2.8 we have generated a sequence of phone-grams for each language. We have tested several options for the feature vector size, obtaining an optimum for size 40, being 240 the final vector, considering the 6 languages to be recognized. In relation to the number of Gaussians in the i-Vectors system we have obtained an optimum for 512. The best result was 24.69% of Cavg.

3.2. MVE using the Skip-Gram model

When we considered a single Phone-based Embedding feature vector for each language as in SVE we obtained 29.15% of Cavg using only the model for the Basque language. After fusing all the languages we obtained 19.73% of Cavg, which is a relative improvement of 20.1% over SVE. So, we decide to use MVE in all remaining experiments.

3.3. Inclusion of Context in the vector embeddings

Based on the hypothesis that the vector representations obtained from NN have some linguistic regularities as the additive composition [13], we propose to use Contextual information, including information from the neighboring phone-grams in the final Phone-based Embedding. Our proposal is to use a weighted sum of neighboring phone-grams. We considered two options for that, the first one...
uses a three phone-gram context window and the second one a five phone-gram context window with the following weights:

A) 3 phone-grams context: Final Ph-Emb = Left Ph-Emb * 0.25 + Central Ph-Emb * 0.50 + Right Ph-Emb * 0.25

B) 5 phone-grams context: Final Ph-Emb = Second Left Ph-Emb * 0.10 + Left Ph-Emb * 0.15 + Central Ph-Emb * 0.50 + Right Ph-Emb * 0.15 + Second Right Ph-Emb * 0.10

The objective is to assign more weight to the current phone-gram but taking into account information from the neighboring units. Using option A (Cavg: 24.38) we obtained a 14.4% relative improvement over option B (Cavg: 28.49) using the model for the Basque language and the Skip-Gram technique as reference. The optimum vector size for SG-Emb is 80, with 512 Gaussians in the iVectors system, 10 iterations and a window size of 8. All the optimization has been obtained using the data development set. After fusing all the languages, we obtained 18.70% of Cavg with MVE, which is a relative improvement of 24.3% over SVE.

3.4. GloVe model for the MVE

We have also evaluated our approach using the GloVe model (Section 2.6) instead of the Skip-Gram model for our best system with contextual information, because it incorporates information of the co-occurrence of phone-grams in all the training data set. The optimal configuration parameters are: vector size of 80, window size of 4, and 30 iterations. The optimum number of Gaussians for the iVectors system has been 512 (the same as Skip-Gram). Fusing all the languages as before, we obtain a 16.70% of Cavg, which is a 10.7% of relative improvement over MVE based on the Skip-Gram model.

3.5. Summary of results

In Table 2 we present the summary of results obtained with the techniques proposed in this paper. As we can see, the final system using the GloVe model provides the best results.

**Table 2: Summary of results.**

<table>
<thead>
<tr>
<th>System</th>
<th>Cavg</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVE</td>
<td>24.69</td>
<td></td>
</tr>
<tr>
<td>MVE context and Skip-Gram</td>
<td>18.70</td>
<td>24.3</td>
</tr>
<tr>
<td>MVE context and GloVe</td>
<td>16.70</td>
<td>32.4</td>
</tr>
</tbody>
</table>

3.6. Fusion with the acoustic model

The objective of this technique is to improve an existing LID system, which is based on acoustic information (section 2.10). So, we present the results of fusing the existing acoustic LID system with our two best systems, based on Phone-based Embeddings obtained with the Skip-Gram and Glove models (Table 3).

**Table 3: Phone-based Embeddings systems fused with an acoustic system.**

<table>
<thead>
<tr>
<th>System</th>
<th>Cavg</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acoustic system</td>
<td>7.60</td>
<td></td>
</tr>
<tr>
<td>Fusion with SG-Emb</td>
<td>5.40</td>
<td>28.9</td>
</tr>
<tr>
<td>Fusion with Glo-Emb</td>
<td>5.01</td>
<td>34.1</td>
</tr>
</tbody>
</table>

4. Conclusions

We have demonstrated that the use of Phone-based Embeddings as feature vectors provides improvements in an LID task. We have used as a baseline a first system that uses Phone-based Embeddings as feature vectors with rather poor results. However, using the new approaches proposed in this paper results improved, and the fusion of our best configuration with an acoustic system provides significant improvements.

Our baseline system uses the "SVE" technique to obtain 24.69% of Cavg. Considering this poor result, we decided to change the approach and use an individual matrix for each language, fusing the scores from all individual systems at the back-end, and we obtained 19.73% of Cavg using the Skip-Gram modelling.

Then, we proposed the inclusion of context information in the Phone-based Embeddings including the two or four nearest neighbours. After fusing all the language models we obtained 18.7% of Cavg using the Skip-Gram modelling. Finally, using the GloVe modelling we obtain 16.7% of Cavg with a 10.7% relative improvement over Skip-Gram modelling and a 32.4% compared to the baseline system.

Also, the fusion with the acoustic based system provides a 34.1% relative improvement, which demonstrates that both systems provide complementary information for the LID task.

As future research lines, we propose to study the effects of higher order units using a larger database. We will also evaluate other types of language models for the neural embeddings. Also, we expect to use models with a high number of layers (char-RNN) and use its combination with convolutional DNNs to get better local context characteristics.

5. Acknowledgements

The work leading to these results has been supported by AMIC (MINECO, TIN2017-85854-C4-1-R), and CAVIAR (MINECO, TEC2017-84593-C2-1-R) projects. Authors also thank Mark Hallet for the English revision of this paper and all the other members of Speech Technology Group for the continuous and fruitful discussion on these topics. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

6. References


End-to-End Speech Translation with the Transformer

Laura Cros Vila, Carlos Escolano, José A. R. Fonollosa, Marta R. Costa-jussà
TALP Research Center, Universitat Politècnica de Catalunya, Barcelona
lcrosvila@gmail.com, {carlos.escolano, jose.fonollosa, marta.ruiz}@upc.edu

Abstract

Speech Translation has been traditionally addressed with the concatenation of two tasks: Speech Recognition and Machine Translation. This approach has the main drawback that errors are concatenated. Recently, neural approaches to Speech Recognition and Machine Translation have made possible facing the task by means of an End-to-End Speech Translation architecture.

In this paper, we propose to use the architecture of the Transformer which is based solely on attention-based mechanisms to address the End-to-End Speech Translation system. As a contrastive architecture, we use the same Transformer to build the Speech Recognition and Machine Translation systems to perform Speech Translation through concatenation of systems.

Results on a Spanish-to-English standard task show that the end-to-end architecture is able to outperform the concatenated systems by half point BLEU.

Index Terms: End-to-End Speech Translation, Transformer

1. Introduction

The fields of Machine Translation (MT) and Automatic Speech Recognition (ASR) share many features, including conceptual foundations, sustained interest and attention of researchers in the field, a remarkable progress in the last two decades and the resulting wide popular use. Both ASR and MT have a long way to improve and, as a result, do not give perfect results. Speech Translation (ST) applications are typically created by combining ASR and MT systems [1, 2].

This pipeline implies that each system has to be trained with their own dataset (which are required to be large) creating a big drawback for low resourced languages. In addition, all errors made by the recognizer go to the MT system and then the MT system itself adds its own errors. The errors are combined, and the results are often very poor.

Deep learning architectures have allowed for end-to-end approaches for both machine translation [3] and speech recognition [4]. Both systems are based on an architecture of encoder-decoder with recurrent neural networks and attention mechanisms. This architecture has been successfully extended to end-to-end speech translation [5].

Recently, there has been a new proposed architecture for addressing machine translation [6]. Later, this architecture has also been used for speech recognition7 [7]. In both cases, the Transformer outperforms previous architectures based on recurrent neural networks. Inspired by these previous works, this paper describes how to adapt this architecture to end-to-end speech translation. The rest of the paper is organised as follows. Section 2 briefly describes the architecture of the Transformer to make this paper self-contained. Section 3 reports the details on the experimental part including databases, training parameters and results. Finally, 4 reports the final conclusions.

2. Transformer

The goal of this work is to build an End-to-End Speech Translation (ST) based on the Transformer [8]. This end-to-end task will be contrasted with the concatenation of the Automatic Speech Recognition (ASR) and Machine Translation (MT) systems, which are also built using the Transformer architecture. This section briefly describes this architecture.

As many neural sequence transduction models, the Transformer has an encoder-decoder structure. The main difference between it and any other model is that Transformer is entirely based on attention mechanisms [9] and point-wise, fully connected layers for both the encoder and the decoder. This makes it computationally cheaper than other architectures with similar test scores. The whole architecture of the Transformer is depicted in (Figure 1).

Figure 1: Model architecture of the Transformer.

2.1. Input/Output

Originally, the input of the Transformer is a sequence of words divided in sub-units denominated tokens. Once the text is turned into a tokenized version of the words, a matrix of real numbers collects the vectors (typically of size $d_{model} = 512$). If taken the raw input sequence as $x = (x_1, x_2, \ldots, x_m)$ and the embedded representation as $w = (w_1, w_2, \ldots, w_m)$ with $w_j \in \mathbb{R}^V$, then each $w_j$ is a column vector of the input matrix belonging to the space $\mathbb{R}^{V \times f}$, with $V$ as the number of embeddings and $f$ the number of features of each embedding.

The decoder generates an output sequence corresponding to the

---

7 https://tensorflow.github.io/tensor2tensor/tutorials/asr

---

IberSPEECH 2018
21-23 November 2018, Barcelona, Spain
input sentence.

2.2. Positional encoding
The lack of recurrence and convolution in the model entails that no recurrence nor temporal information is available. A good way to keep the order of the sentence is adding positional encoding to the input embeddings at the bottoms of the encoder and decoder stacks.

What is used in Transformer is an element-wise vector \( p = (p_1, p_2, \ldots, p_n) \), with \( p_j \in \mathbb{R}^d \), which is added to the original matrix.

2.3. Encoder
The encoder consists of a stack of \( N \) layers, each of them composed of two sub-layers: a multi-head attention mechanism and a fully-connected feed forward net, plus residual connections \( 2 \) (referred to in Figure 1 as "Add") on both stages, followed by a layer of normalization.

The multi-head attention has several parallel attention layers, or heads, which concatenate attention functions with different linearly projected queries, keys and values.

2.4. Decoder
The decoder resembles the encoder but it is not completely equal. Although it is also a stack of \( N \) layers with sub-layers within them, a masked multi-head attention layer is added (apart from the common residual connections and layer normalization). The singular fact of the decoder is that at each step the model is auto-regressive, meaning that it uses the previously generated symbols as additional input when generating the next ones.

3. Experimental work
Our implementation of the Transformer is based on Tensor2Tensor [8], or T2T for short, which is a library of deep learning models and datasets actively used and maintained by researchers and engineers within the Google Brain team and a community of users. As follows, we report the database used, the parameters to train the system and the results.

3.1. Database
The database used in this experiment is the Fisher Spanish and Callhome Spanish Corpus. The Fisher Spanish Corpus provides a set of speech and transcripts developed by the Linguistic Data Consortium (LDC) which consists of audio files covering roughly 163 hours of telephone speech from 136 native Caribbean Spanish and non-Caribbean Spanish speakers. The speech recordings consist of 819 telephone conversations of 10 to 12 minutes in duration. Full orthographic transcripts of these audio files are available in "LDC2010T04. The audio files are available in "LDC2010T04.

The CALLHOME Spanish Corpus consists of 120 unscripted telephone conversations between native speakers of Spanish. All calls, which lasted up to 30 minutes, originated in North America and were placed to international locations. Most participants called family members or close friends. The audio files of the CALLHOME Corpus are available at "LDC96S35. The transcripts of these audio files are available at "LDC96T17. The transcript files are in plain-text, tab-delimited format (tfd) with UTF-8 character encoding. In order to adapt the transcript files for the Transformer, all the text was turned into capital letters as well as a reference number at the beginning of each sentence was added, consisting of six digits starting from "000000" to the last sentence and separated with a tabulator such as follows:

000000 HELLO
000001 ALO.
000002 ALO, BUENAS NOCHES. QUÉN ES?
000003 QUÉ TAL, EH, YO SOY GUILLERMO, COÑO
ESTÁS?
000004 AH GUILLERMO.
...
003637 OH MY GOD.
003638 MHM, Y NO LE PODÍAN HACER NADA, NO.
003639 MM.

The conversations were recorded as 2-channel mu-law sample data with 8000 samples per second (as captured from the public telephone network).

Table 1 shows the corpus statistics of the text dataset for English-Spanish.

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>S</th>
<th>W</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanish</td>
<td>138819</td>
<td>1503003</td>
<td>50857</td>
<td></td>
</tr>
<tr>
<td>Dev1</td>
<td>3979</td>
<td>41271</td>
<td>6251</td>
<td></td>
</tr>
<tr>
<td>Dev2</td>
<td>3960</td>
<td>40072</td>
<td>5793</td>
<td></td>
</tr>
<tr>
<td>Test</td>
<td>3641</td>
<td>40141</td>
<td>5888</td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>138819</td>
<td>1441090</td>
<td>37817</td>
<td></td>
</tr>
<tr>
<td>Dev1</td>
<td>3979</td>
<td>40015</td>
<td>5553</td>
<td></td>
</tr>
<tr>
<td>Dev2</td>
<td>3977</td>
<td>39799</td>
<td>5109</td>
<td></td>
</tr>
<tr>
<td>Test</td>
<td>3641</td>
<td>39152</td>
<td>4739</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3641</td>
<td>39513</td>
<td>4734</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3641</td>
<td>39004</td>
<td>4840</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3641</td>
<td>39617</td>
<td>4929</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3641</td>
<td>39011</td>
<td>4901</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3641</td>
<td>38578</td>
<td>4875</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Corpus Statistics. Language (L), number of sentences (S), words (W), vocabulary (V).

For the MT experiment, we used the parallel text from LDC2014T23 available from LDC and in github \(^6\).

3.2. Parameters
For the experiments, we used three different architectures. These three architectures correspond to the ASR, MT and ST models.

The main hyperparameters used are detailed in Table 2. For the ASR/ST models the learning rate had a decay each

\(^2\)Assuming that the function to be modeled (weights and bias of the net) is closer to an identity mapping than to a zero mapping, a good way to optimize the learning process is to add residual connections, which provides the input without any transformation to the output of the layer.

\(^3\)https://catalog.ldc.upenn.edu/LDC2010T04

\(^4\)https://catalog.ldc.upenn.edu/LDC96S35

\(^5\)https://catalog.ldc.upenn.edu/LDC96T17

\(^6\)https://github.com/joshua-decoder/fisher-callhome-corpus
5000 steps and the learning rate warm-up steps set to 8000\(^9\). To adapt the speech features in the ASR encoder, we used the `conv.relu.conv` from tensor2tensor. As parameters, we used `mel.filterbank` of 80 coefficients every 10ms with a window of 25 ms. As preprocessing for the ASR inputs, we used the tensor2tensor options as follows `conv1d(inputs, filter_size=1536, kernel_size=9) + relu + conv1d(inputs, filter_size=384, kernel_size=1). The Transformer gets a vector of dimension 384 every 10ms. Also for the speech part, a clarification of the input and target maximum sequence length is that to have an input maximum sequence length of 1550 means that only examples of transcriptions whose audio has less than 1550 frames are used, which implies that with frames of 10 ms the maximum size of the input audio frame is approximately 15.5 seconds in length. On the other hand, to have a target maximum sequence length of 350 means that the train transcriptions are limited to a maximum size of 350 characters.

The speech models were trained on TPUs \([10]\) following the suggested parameters for the librispeech task of the tensor2tensor library \([11]\).

3.3. Training

When training, as there are several GPUs or TPUs, the parameters are applied to each one. As the effective batch size is the numbers of GPUs (in this case 4 or 8 in a TPU) multiplied by the batch size. In each batch the parameters are updated using the stochastic gradient descend and the Adam optimizer \([12]\).

Both the ASR/ST and MT systems use a character-based tokenization. This implies that the models look for the correlation of input and output sentences character by character.

3.4. Results

The evaluation of each model involving translation was done by computing the Bilingual Evaluation Understudy score (BLEU) \([13]\). The BLEU score is the most used for the field of MT and it compares the decoded sentence with the target sentence of the test set by looking into the modified n-gram precision. As for ASR evaluation, a commonly used metric is Word Error Rate (WER) \([14]\), which is defined as the ratio of word errors (substitutions, deletions and insertions) to words processed. For the evaluation of ASR systems, punctuation marks were not taken into account.

Comparing ASR+MT concatenation and End-to-End Speech Translation, the results show that in terms of BLEU, the latter is slightly better than the former gaining 0.5 points of BLEU.

Figure 2 shows an example that when concatenating ASR and MT, the errors are also concatenated. The Spanish target word **BAILA** (**DANCE** in English), when recognized with the model ASR ES, is misspelled and transcribed to the word **VAYA**, which has a very similar sound but totally different meaning. As a consequence, the final translation output can not reproduce the word **DANCE**, which gives a strong meaning of context to the sentence. In this case, the end-to-end system is able to produce a better translation.

4. Conclusions

This paper proposes to use of the Transformer as main architecture for Speech Recognition, Machine Translation and Speech Translation. To the best of our knowledge, this is the first time that this promising architecture is used to reproduce an End-to-End Speech Translation system. BLEU results show that the End-to-End Speech Translation architecture provides slightly better results than the standard ASR and MT concatenation. Examples show that these better results are achieved by avoiding the concatenation of errors.

In future work, it would be interesting to train a system capable of doing multi-task learning \([15]\). This system would build several models and not only one learning to translate from Spanish speech to English text. The new multi-task model would learn in addition Spanish Recognition and/or Spanish-to-English text translation.

5. Acknowledgements

This work is supported in part by the Spanish Ministerio de Economía y Competitividad, the European Regional Development Fund and the Agencia Estatal de Investigación, through the postdoctoral senior grant Ramón y Cajal, the contract TEC2015-69266-P (MINECO/FEDER,EU) and the contract PCIN-2017-079 (AEI/MINECO).

6. References


<table>
<thead>
<tr>
<th>Hparam</th>
<th>Text-to-Text (GPU)</th>
<th>ASR/ST (TPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of encoder layers</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Number of decoder layers</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Gradient clipping</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.2</td>
<td>0.15</td>
</tr>
<tr>
<td>Momentum</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>Audio sampling rate</td>
<td>-</td>
<td>8000</td>
</tr>
<tr>
<td>Batch size</td>
<td>4096</td>
<td>16</td>
</tr>
<tr>
<td>Maximum length</td>
<td>256</td>
<td>125550</td>
</tr>
<tr>
<td>Input sequence maximum length</td>
<td>0</td>
<td>1550</td>
</tr>
<tr>
<td>Target sequence maximum length</td>
<td>0</td>
<td>350</td>
</tr>
<tr>
<td>Adam optimizer</td>
<td>$\beta_1 = 0.9 \beta_2 = 0.997 \epsilon = 10^{-7}$</td>
<td>$\beta_1 = 0.9 \beta_2 = 0.997 \epsilon = 10^{-7}$</td>
</tr>
<tr>
<td>Attention layers</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>Initializer</td>
<td>uniform unit scaling</td>
<td>uniform unit scaling</td>
</tr>
<tr>
<td>Initializer gain</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Training steps</td>
<td>250000</td>
<td>210000</td>
</tr>
</tbody>
</table>

Table 2: Training parameters.

<table>
<thead>
<tr>
<th>System</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR ES (TPU)</td>
<td>38.02 (WER)</td>
</tr>
<tr>
<td>MT (GPU)</td>
<td>55.05 (BLEU)</td>
</tr>
<tr>
<td>ASR + MT (TPU/GPU)</td>
<td>19.97 (BLEU)</td>
</tr>
<tr>
<td>ASR EN (TPU)</td>
<td>20.47 (BLEU)</td>
</tr>
</tbody>
</table>

Table 3: Results of the model evaluation. ASR ES stands for the speech recognition with Spanish transcriptions as target. ASR EN stands for the speech recognition with English transcriptions as target.


63
Audio event detection on Google's Audio Set database: Preliminary results using different types of DNNs

Javier Darna Sequeiros and Doroteo T. Toledano

AuDlas – Audio, Data Intelligence and Speech, Universidad Autónoma de Madrid
javier.darna@estudiante.uam.es, doroteo.torre@uam.es

Abstract

This paper focuses on the audio event detection problem, in particular on Google Audio Set, a database published in 2017 whose size and breadth are unprecedented for this problem. In order to explore the possibilities of this dataset, several classifiers based on different types of deep neural networks were designed, implemented and evaluated to check the impact of factors such as the architecture of the network, the number of layers and the codification of the data in the performance of the models. From all the classifiers tested, the LSTM neural network showed the best results with a mean average precision of 0.26652 and a mean recall of 0.30698. This result is particularly relevant since we use the embeddings provided by Google as input to the DNNs, which are sequences of at most 10 feature vectors and therefore limit the sequence modelling capabilities of LSTMs.

Index Terms: Audio Set, deep neural network, audio event recognition, machine learning.

1. Introduction

In machine learning, there are several problems that try to mimic biologic senses, such as recognizing objects in images (image object recognition), or identifying particular sounds in an audio track (audio event recognition). In recent years, there have been great improvements in image object recognition thanks to the availability of large databases such as the Imagenet database [1] and similar ones. These have allowed the organization of competitions and have fostered the proposal of novel network architectures such as the Alexnet, VGG, Residual Networks, etc. In the case of audio event recognition, the lack of availability of large databases has not allowed a similar development until very recently. There have been several notable efforts to foster research in this area, among which we must mention:

- The CLEAR audio event recognition and classification challenge [2], which compared algorithms on a database with 12 audio event classes including common sounds from meeting rooms and seminars.
- The urban sound taxonomy [3] dataset, containing 10 classes of urban sounds.

Since 2013, the Detection and Classification of Acoustic Scenes and Events (DCASE) community (http://dcase.community/) has organized several challenges focused on different acoustic scenes and events detection and classification problems. Since 2016, there are yearly challenges and workshops. The 2018 edition [4] received 223 submission entries from 81 teams, which implies a huge collective effort in the area. It proposed five different tasks, one of them being about general-purpose audio tagging. The database used in this task is a subset of the Freesound database [5], a collaborative database of sounds under a Creative Commons license. This subset has 11.1K segments and 41 different categories taken from the Google Audio Set ontology.

Along with these databases and challenges, several models based on neural networks were developed in order to classify the data in such databases, quickly becoming state-of-art over previous models mostly based on hidden Markov models (HMM):

- The paper “Polyphonic sound event detection using multi label deep neural networks” [6] uses a neural network for multi label audio event classification that obtained 63.8% accuracy, outperforming the previous state-of-art model based on HMMs.
- The paper “Recurrent Neural Networks for Polyphonic Sound Event Detection in Real Life Recordings” [7] proposes a bi-directional LSTM network to classify audio events from a database with 61 classes from 10 different contexts. This system reports an average F1-score of 65.5%.

Nevertheless, these studies focus on somewhat restricted tasks, and the databases used are relatively small, in terms of both number of samples and number of classes, which seriously limits the applicability of modern deep learning techniques and the progress made in audio event recognition for general cases.

For those reasons, Google created a database named Google Audio Set consisting of segments of 10 seconds extracted from YouTube videos, and published it in 2017 [8]. With over 2 million samples and 527 different classes, this database is significantly larger and wider than any other database ever created for this problem, thus allowing to train and test more versatile models. In particular, such a huge database is very well fitted to the problem of deep learning where very complex models can be trained from huge amounts of data. In addition to the database itself, Google trained a model on it in order to establish a baseline. This model, described as a multilayer perceptron with a single hidden layer of units (hence not a deep neural network) produced a mean average precision (mAP) of 0.314 [9] on the evaluation subset. This value can be used as a baseline for other models, but unfortunately, Google did not publish more details on this system. As the database was published just one year before the moment this article was written, there has not been great improvement from the previously mentioned baseline. Despite that, the current state-of-art results have been achieved through the use of attention models, reaching mAPs of 0.327 with the inclusion of a trainable probability measure for the
samples [10], and 0.360 with the implementation of several levels of attention [11]. This paper intends to be a first approach to Google Audio Set. Our goal is to train several different architectures of neural networks with this database and compare their evaluation results with each other and with Google's baseline. The rest of the paper is organized as follows: Section 2 will discuss in more detail the Google Audio Set database. Section 3 will define the neural networks that were used to face the audio event detection problem, section 4 will describe the tests that were made on the neural networks created, section 5 will interpret the results from those tests, and finally section 6 will conclude the work and propose future research lines that arise from the results of this paper.

2. The Google Audio Set database

The Google Audio Set database is available in two different formats:

- Text files describing the video id, start time, stop time, and labels assigned to each segment.
- Features (embeddings) extracted at 1 Hz for each segment using a DNN trained by Google (the structure of this DNN is similar to the VGG networks used in image recognition).

In this paper, we will use only the latter format so that minimal preprocessing is required and results are easier to recreate. However, the official dataset from Google Audio Set webpage was not used because it was developed to be used directly on Tensorflow (also developed by Google) and it was very difficult and inefficient to use in other toolkits such as Keras. Instead, we finally used an “unofficial” conversion to .h5df format available from the Google+ user group “audioset-users” [12]. This conversion includes, for each segment, the extracted audio features as a uniform 128x10 array and the presence or absence of each possible label as a boolean vector.

The Google Audio Set database consists of 3 subsets (in any of the available formats):

- Balanced training, which has a balanced distribution of the classes but contains only a small fraction of the samples (about 22K segments).
- Unbalanced training, which contains all of the samples (2.0M segments) but suffers from a greatly unbalanced distribution of the classes.
- Evaluation, which contains about 20K segments.

In all the experiments of this paper, the neural networks were trained with one of the training sets and then tested with the evaluation set in order to obtain the final results.

3. Proposed neural networks

All the models used in this paper to perform the test are neural networks based on one of three different architectures. Despite the different architecture, all the models share the following properties:

- They use Adam as their optimizer.
- Every unit that does not belong to the output layer uses ReLU as its activation function.
- The output layer is fully connected and has 527 units.

Since the audio event classification is a multi-labelled problem (i.e. the same segment can, and typically, contain more than one type of audio), all the networks (except one, as will be discussed later) use binary cross-entropy as their loss function and the binary sigmoid as the activation function for the output layer.

3.1. LSTM.

Since the data has the form of a time series of 10 feature vectors (embeddings), each one representing one second of audio, a Long Short-Term Memory (LSTM) recurrent neural network was immediately considered as an appropriate network, as they are specialized in this kind of data. The proposed LSTM model has the following architecture:

- The inputs are series of 10 128-dimensional feature vectors (embeddings).
- The first hidden layer is a unidirectional LSTM layer with 600 units, which outputs a single vector when the whole sequence has been processed. This layer is followed by a dropout layer with a 0.3 probability.
- The last hidden layer is a fully connected layer with 600 units.

3.2. CNN.

Convolutional Neural Networks (CNN) have proven to be very powerful for processing images. In our case, the input sequence of 10 128-dimensional feature vectors can be considered as a 10x128 image or matrix, and therefore a CNN could be appropriate in this case as well. The 128-dimensional input vectors are themselves produced as the output of a different neural network. This neural network uses a PCA transformation to create an embedding of the data. Therefore, there is no local proximity relationship between the elements of the vector. However, there is a temporal proximity for each individual feature, which translates into a local proximity between the rows of the matrix. A convolutional neural network with the following architecture was designed to take advantage of this temporal proximity:

- The input is the previously mentioned 10x128 matrix.
- The first hidden layer is a convolutional layer with 16 filters and a kernel with dimension 3x1. Since there is no proximity relationship in the input vectors (columns of the input matrix) the second dimension of the kernel is always restricted to 1 in our tests.
- The second hidden layer is a maximum pooling layer with a 2x2-dimensional window followed by a dropout layer with a 0.3 probability.
- The last hidden layer is a fully connected layer with 600 units.

3.3. MLP

Multi Layer Perceptrons (MLPs) are amongst the most standard and versatile neural networks. In fact, MLPs can approximate any input-output multidimensional output, including those produced by other network architectures, so they can be used as a reference model. The main advantage of other architectures over MLPs is that MLPs include a huge amount of weights, which can make training more difficult and more prone to overfitting. Given their property of
universal function approximation, they can be used to test the impact of other factors apart from the type of neural network they are based on. For this last reason, several models based on MLPs were developed and tested:

- Two MLPs with one hidden layer.
- A MLP with two hidden layers.
- A MLP with three hidden layers.

All these models share these properties:

- The input is a 1280-dimensional vector (the flattened version of the input matrix used for CNNs).
- The hidden layers have 1500 units each. After each one of them, there is a dropout layer with a 0.3 probability.

One of the MLPs with one hidden layer has the following particular properties:

- The hyperbolic tangent (tanh) is used as the activation function of the output layer.
- In this case, the Mean Squared Error (MSE) is used as the networks' loss function.

4. Test Description

In order to compare the performance of the different models, we evaluated them on the evaluation subset of Google Audio Set. Every model was trained with the balanced training test of Audio Set. In the case of the MLP with the bipolar sigmoid activation function, the target vectors were preprocessed so that they have a bipolar codification (the absence of a class is represented with the value -1 instead of 0), allowing us to test the effect of the codification of the data in performance.

All the models were trained with a minibatch size of 128. The training had a maximum duration of 50 epochs, however, early stopping was used in order to interrupt the training process when the mAP no longer increases for three epochs, thus preventing overfitting. No early stopping was used on the LSTM model as its mAP grew at a notably irregular rate and early stopping kept interrupting the training process before the model could reach its stability phase. This phenomenon didn’t happen with the rest of the models.

As the main focus of these tests is to compare the different network architectures, hyper-parameters were left at their default values (learning rate: 0.001, beta1: 0.9, beta2: 0.999, decay: 0).

After training, the networks were tested with the evaluation subset of the Google Audio Set, and the final results were obtained by calculating the mean Average Precision (mAP) and the mean Recall (mR).

In addition, another test was performed on all the models based on MLPs where they were trained with the unbalanced training set (much larger in terms of samples, but much more unbalanced too) instead of the balanced one.

5. Results

After performing the tests described above, results presented in Table 1 were obtained:

<table>
<thead>
<tr>
<th>Model, training set, codification</th>
<th>mAP</th>
<th>mR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP 1 h. l., bal., bip.</td>
<td>0.13704</td>
<td>0.15848</td>
</tr>
<tr>
<td>MLP 1 h. l., unbal., bip.</td>
<td>0.19696</td>
<td>0.22697</td>
</tr>
<tr>
<td>MLP 3 h. l., unbal., bin.</td>
<td>0.20686</td>
<td>0.23529</td>
</tr>
<tr>
<td>MLP 2 h. l., unbal., bin.</td>
<td>0.21203</td>
<td>0.24079</td>
</tr>
<tr>
<td>MLP 1 h. l., unbal., bin.</td>
<td>0.21342</td>
<td>0.24166</td>
</tr>
<tr>
<td>MLP 1 h. l., bal., bin.</td>
<td>0.21893</td>
<td>0.24249</td>
</tr>
<tr>
<td>CNN, bal., bin.</td>
<td>0.22830</td>
<td>0.25595</td>
</tr>
<tr>
<td>MLP 2 h. l., bal., bin.</td>
<td>0.24422</td>
<td>0.27542</td>
</tr>
<tr>
<td>MLP 3 h. l., bal., bin.</td>
<td>0.25276</td>
<td>0.28706</td>
</tr>
<tr>
<td>LSTM, bal., bin.</td>
<td>0.26652</td>
<td>0.30698</td>
</tr>
</tbody>
</table>

The first point to note is that the ranking of the different neural networks is the same whether the models are ordered by their mAP or their mR. Because of that, the metric considered is irrelevant when interpreting the results.

The models using bipolar data obtain the worst results. One possible reason could be that hidden layers use an activation function unable to take negative values. However, by comparing both models using bipolar data, we can notice that the one using the unbalanced training set has a much better performance than the one using the balanced training set. This is surprising because the models using binary data show the exact opposite behavior. In these models the use of the unbalanced training set has a negative impact on their performance, the effect becoming more intense the more layers the network has.

The MLPs using binary data and the balanced training set have a better performance the more hidden layers they have, which is the expected behavior when there is enough training data, as it seems to be the case.

The CNN's results were quite limited, falling behind the MLPs with more than one hidden layer. This is probably because of the lack of local meaning of the different features included in the 128-dimensional feature vectors (embeddings), which limits the kernel to a single dimension.

Finally, the model with the best results is the LSTM network, with a mAP of 0.26652 and a mR of 0.30698. This result is interesting because, even knowing that LSTM neural networks are particularly effective with time series, in our case these time series are very short, with 10 elements, which could have limited the performance of this model.
6. Conclusions and future work
After testing the performance of several deep neural networks, we were able to obtain a mAP of 0.26652 with a simple LSTM network. Despite these results being worse than the 0.314 mAP of the baseline established by Google, they allow us to draw some conclusions about creating models for Google Audio Set.

First of all, we can conclude that LSTM networks are the most appropriate architecture for this problem from all of those which were tested, as a relatively simple network with one LSTM layer and a fully connected layer offered better results than a more complex network with three fully connected layers, therefore recurrent neural networks should be a good starting point if a better performance is looked for, for example by adding more layers to the model or implementing more complex architectures.

The use of the balanced training subset seems to improve the performance of the models despite being less than 1/20 of the dataset. However, the unbalanced training subset was only used on MLPs due to time restrictions. Its effects on the other architectures should be studied in the future.

Transforming the target vectors to a bipolar codification decreases the performance of the models; however there seems to be a positive correlation between this codification and the use of the unbalanced training set, which could be worth researching into.

7. Acknowledgements
This work has been partially supported by project “DSSL” (TEC2015-68172-C2-1-P), funded by the Ministry of Economy and Competitiveness of Spain and FEDER.

8. References
Emotion Detection from Speech and Text

Mikel deVelasco, Raquel Justo, Josu Antón, Mikel Carrilero, M. Inés Torres
Universidad del Pais Vasco UPV/EHU
mikel.develasco@ehu.eus, raquel.justo@ehu.eus, josuantonsanz@gmail.com, mcarrilero001@ikasle.ehu.eus, manes.torres@ehu.eus

Abstract

The main goal of this work is to carry out automatic emotion detection from speech by using both acoustic and textual information. For doing that a set of audios were extracted from a TV show where different guests discuss about topics of current interest. The selected audios were transcribed and annotated in terms of emotional status using a crowdsourcing platform. A 3-dimensional model was used to define an specific emotional status in order to pick up the nuances in what the speaker is expressing instead of being restricted to a predefined set of discrete categories. Different sets of acoustic parameters were considered to obtain the input vectors for a neural network. To represent each sequence of words, a models based on word embeddings was used. Different deep learning architectures were tested providing promising results, although having a corpus of a limited size.

Index Terms: Emotion Detection, Speech, Text transcriptions

1. Introduction

The emotion recognition is the process of identifying human emotions, a task that is automatically carried out by humans considering facial and verbal expressions, body language, etc. However, this is a challenging task for an automatic system. In recent years, the great amount of multimedia information available due to the extensive use of the Internet and social media, and with new computational methodologies related to machine learning, have led to the scientific community to put a great effort in this area [1, 2].

Emotion recognition from speech signals relies on a number of short-term features such as pitch, vocal tract features such as formants, prosodic features such as pitch loudness, speaking rate, etc. Surveys on databases, classifiers, features and classes to be defined in the analysis of emotional speech can also be found in [3]. Regarding methodology, statistical analysis of feature distributions has been traditionally carried out. Classical classifiers such as the Bayesian or Super Vector Machines (SVM) have been proposed for emotion features from speech. The model of continuous affective dimensions is also an emerging challenge when dealing with continuous rating emotion labelled during real interaction [1, 4]. In this work, recurrent neural networks have been proposed to integrate contextual information and then predict emotion in continuous time using a three-dimensional emotional model.

Speech transcripts have also been demonstrated to be a powerful tool to identify emotional states [5]. Over the last decade, there has been considerable work in sentiment analysis [6]. Moreover, the detection of emotions such as anger, joy, sadness, fear, surprise, and disgust have also been addressed [7]. However, spoken language is informal and provides information in an unstructured way so that developing tools to select and analyse sentiments, opinions, etc. is still a challenging topic [8]. In early systems dealing with emotion detection in text, knowledge-based approaches were applied making use of emotion lexicons, such as Sentiwordnet [9]. Other methods, employ machine learning based approaches [10], where statistical classifiers are trained using large annotated corpora and the emotion detection can be seen as a multi-label classification problem. In this work we propose to use neural networks to solve the regression problem given the 3-dimensional emotional model.

The main contribution of this work is the appropriate selection of a neural network architecture for emotion detection considering both acoustic signals and their corresponding transcription. Additionally, the proposed architecture has been adapted to the 3-dimensional VAD (Valence, Arousal and Dominance) emotional model [11].

Section 2 describes the two proposed approaches for the automatic emotion recognition from used features to the common network architectures basics. Experiments carried out are fully described in Section 3. Section 3.1 aims to explain the difficulties to find out a Spanish corpus and it has led us to create our own corpus. Section 3.2 mentions the used baselines methods and which measure has been used for testing. Section 3.3 shows the experiments carried out under the regression problem of emotional status with acoustic features whereas Section 3.4 deals with the experiments achieved at the regression problem of emotional status with language features. Finally some concluding remarks are reported in Section 4.

2. Emotion Detection from Speech and Language

Emotion detection from speech is based on the extraction of relevant features from the acoustic signal, that can be seen as hints of the emotional status. That is, a numerical vector that represent the specific information related to emotional status and embedded in an acoustic signal is needed. There are numerous acoustic parameters that can be obtained using the free software Praat\(^1\) or the free python library pyAudioAnalysis\(^2\). Considering these tools the following set of 72 parameters could be considered: Pitch, Zero Crossing Rate (ZCR), Energy, Entropy of the energy, Spectral Centroid, Spectral Spread, Spectral Entropy, Spectral Flux, Spectral Rolloff, Chroma vector (12 coefficients), Chroma deviation, MFCC coefficients (12), LPC coefficients (16), Bark features (21). However, some of these parameters provide the same or very similar kind of information. For instance, LPC, Bark and MFCC coefficients provide similar information about the phonemes without considering the vocal tract. Thus, such a big set of parameters is useless since it will complicate the learning procedure requiring more training data. In this work different subsets of the aforementioned parameters were explored:

\(^1\)http://www.fon.hum.uva.nl/praat/
\(^2\)https://pypi.org/project/pyAudioAnalysis/
• Set A: Pitch and Energy.
• Set B: Pitch, Energy and Spectral Centroid.
• Set C: Pitch, Energy, Spectral Centroid, ZCR and Spectral Spread.
• Set E: Pitch, Energy, Spectral Centroid, ZCR, Spectral Spread and 16 LPC coefficients.

The first set was selected according to the studies performed in [12] where the arousal state of the speaker affects the overall energy and pitch. In addition to time-dependent acoustic features such as pitch and energy, spectral features were selected for Sets B and C as a short-time representation for speech signal [13]. For Sets D, E and F different Cepstral-based features were added, proven that they are good for detecting stress in speech signal [14].

When regarding emotion detection from language the same procedure has to be carried out. First of all a vectorial representation of the transcribed text is needed. In this case, we hope to capture some meaning of the utterance that might help in the detection of specific emotional status. An appropriate representation should consider some semantic information like the word embeddings word2vec [15], doc2vec [16] or GLOVE [17]. Word2vec embeddings, the most simple model, are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space. However, this technique represents each word of the vocabulary by a distinct vector, without parameter sharing. In particular, they ignore the internal structure of words, which is an important limitation for morphologically rich languages. For example, in Spanish, most verbs have more than forty different inflected forms and this leads to a vocabulary where many word forms occur rarely (or not at all) in the training corpus, making it difficult to learn good word representations. Thus, [15] proposes to learn representations for character n-grams, and to represent words as the sum of the n-gram vectors. The model (known as FastText) can be seen as an extension of the continuous skip-gram model [18] which takes into account subword information.

Additionally, a way of representing the emotional status is needed in order to establish a machine learning problem. A categorical emotion description (e.g. six basic emotions) is an easy way to procedure but it provides a quite constrained model. Affective computing researchers have started exploring the dimensional representation of emotion [19] as an alternative. Dimensional emotion recognition, aims to improve the understanding of human affect by modelling affect as a small number of continuously valued, continuous time signals. It has the benefit of being able to: (i) encode small changes in affect over time, and (ii) distinguish between many more subtly different displays of affect, while remaining within the reach of current signal processing and machine learning capabilities [20].

In our work, we represent the problem of dimensional emotion recognition as a regression one, where each emotional status is represented as three-dimensional real-valued vector. The dimensions of this vector correspond to Valence (corresponding to the concept of polarity), Arousal (degree of calmness or excitement), and Dominance (perceived degree of control over a situation): the VAD model.

In order to solve the regression problem of emotional status detection, we propose to use deep learning. When considering emotion detection from speech Long-Short Term Memory (LSTM) neural networks were tested. The underlying idea is to be able to learn the relationship among present and past information although existing a big distance among them. That is, they have memory and they can manage with temporal sequences of data like the sequence of vectors extracted from an acoustic signal. When regarding emotional status detection from text a classical feedforward network was considered because of simplicity. Such networks have proven to be efficient for problems in similar tasks, like sentiment analysis [21].

3. Experiments
We have carried out two series of experiments for the evaluation of the regression processes. In the first one we present the most interesting results related to the emotion detection from speech, and in the second we show the most interesting results on emotion detection from text.

3.1. Corpus
As far as we know there is no Spanish three-dimensional corpus within the literature, so for the experiments, we have created a small corpus using the VAD model. The corpus consists of 120 fragments between 3 and 5 seconds taken from the Spanish TV program “La Sexta Noche”. This TV program consist of political debate, news and events, and discussions commonly appear. Each fragment has been transcribed manually and tagged using crowdsourcing (the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people) techniques. In this case each fragment has been labeled by 5 different annotators, following the next questionnaire:

1. In order to address the Valence: “How do you perceive the speaker?”
   • Excited
   • Slightly Excited
   • Neutral

2. In order to address the Arousal: “His mood is …”
   • Positive (nice / constructive)
   • Slightly Positive
   • Slightly Negative
   • Negative (unpleasant / non collaborative)

3. In order to address the Dominance: “How do you perceive the speaker about the situation in which he or she is in?”
   • More dominant / controlling the situation / …
   • He or she does not dominate the situation neither is he or she cowed.
   • More coward / defensive / …

Once the tags were generated by crowdsourcing, the answers collected from all the annotators were transferred to the three-dimensional model, making the average of each answer...
for all fragments where the first answer of each question was assigned the value 0, the last answer was assigned the value 1, and the rest of the answers a midpoint. Then the corpus was split into two sets, 70% of the fragments were used for training purposes and the remaining 30% for test.

3.2. Baselines models and Evaluation Metrics

Both emotion detections problems, from speech and from text, has been tested first with Linear Regression (LR) [22] and Super Vector Regression (SVR) [23] (with three different types of kernels linear, poly, and rbf), in order to compare with Neural Networks.

Regarding the input of these baselines models, two different approaches have been analysed to fix the problem of time sequence. On the one hand, we fit the models with full information, considering each feature in on each time-step independent (full models), and on the other hand, calculating the mean of each feature over time-steps (mean models).

In relation to the evaluation metric, the Mean Square Error (MSE) has been used, because it seems to provide a good interpretation of how far the prediction and the true label are. In this problem, MSE can be described as the mean of the distances between the points of the true label on the three-dimensional model and the predicted points on the same three-dimensional model.

3.3. Experiments with acoustic features

In order to obtain the acoustic parameters, each audio has been divided into individual frames using a context window of 25 milliseconds and a step of 10 milliseconds (as shown in Figure 1), obtaining 300 frames per audio. A vector made up of the selected acoustic features was associated to each frame. Different experiments were carried out using the different feature sets (A, B, C, D, E, F) described in Section 2. Additionally, for each set, different experiments were also performed including both the first and the first and the second derivatives.

The network proposed to address the regression problem of emotional status with acoustic features is a Recurrent Neural Network (RNN). The network is composed with an LSTM layer of emotional status with acoustic features. Each pair of set and model has been tested with acoustic features itself, with first derivatives and with first and second derivatives, but only best performance is shown. MSE error has been used in order to compare.

As shown in Table 1, the proposed network slightly improves the results of the baseline models in almost all the sets in the corpus. It can also be concluded that by selecting a smaller set of parameters, better results are obtained with the baseline model. However, the set A seems to have insufficient information and Bark features are of great help in the case of networks providing the best results.

3.4. Experiments with language features

Regarding the word representation, FastText embeddings from SBWC has been used in the experiments. The mentioned embeddings are a Skipgram model of 300 dimensions and 855380 different word vectors, trained with Spanish Billion Word Corpus with more than 1.4 billion words.

The regression problem of emotional status with language features has been addressed with a small Deep Neural Network (DNN). This network consist of three similar layers; the first two layers are composed of 5 units, a sigmoidal activation function and followed by Dropout layer with 0.5 of keep-probability in order to prevent the overfitting problem; while the last layer, the output layer, is a Dense layer of 3 units and the sigmoidal activation function (same as the network proposed for the regression problem of emotional status with acoustic features).

As shown in Table 2, the proposed network achieves similar results when comparing it to the baselines models. It is an interesting result given the small size of the training set and the great

![Figure 1: Schematic diagram of speech production.](https://github.com/uchile-nlp/spanish-word-embeddings/blob/master/README.md)

Table 1: Best result obtained with baseline models and Recurrent Neural Network (RNN) in the regression problem of emotional status with acoustic features. Each pair of set and model has been tested with acoustic features itself, with first derivatives and with first and second derivatives, but only best performance is shown. MSE error has been used in order to compare.

<table>
<thead>
<tr>
<th>Set</th>
<th>LR (linear)</th>
<th>SVR (poly)</th>
<th>SVR (rbf)</th>
<th>RNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1682</td>
<td>0.1660</td>
<td>0.1691</td>
<td>0.1670</td>
</tr>
<tr>
<td>B</td>
<td>0.1686</td>
<td>0.1653</td>
<td>0.1685</td>
<td>0.1710</td>
</tr>
<tr>
<td>C</td>
<td>0.1690</td>
<td>0.1679</td>
<td>0.1718</td>
<td>0.1709</td>
</tr>
<tr>
<td>D</td>
<td>0.1742</td>
<td>0.1894</td>
<td>0.1703</td>
<td>0.1710</td>
</tr>
<tr>
<td>E</td>
<td>0.1699</td>
<td>0.2007</td>
<td>0.1842</td>
<td>0.1711</td>
</tr>
<tr>
<td>F</td>
<td>0.1733</td>
<td>0.2256</td>
<td>0.1898</td>
<td>0.1710</td>
</tr>
</tbody>
</table>

Table 2: Best result obtained with baseline models and Deep Neural Network (DNN) in the regression problem of emotional status with language features. MSE error has been used.

<table>
<thead>
<tr>
<th>Set</th>
<th>LR (linear)</th>
<th>SVR (poly)</th>
<th>SVR (rbf)</th>
<th>DNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.1906</td>
<td>0.1530</td>
<td>0.1165</td>
<td>0.1197</td>
</tr>
<tr>
<td>Full</td>
<td>0.1356</td>
<td>0.1292</td>
<td>0.1199</td>
<td>0.1229</td>
</tr>
</tbody>
</table>

As shown in Table 2, the proposed network achieves similar results when comparing it to the baselines models. It is an interesting result given the small size of the training set and the great

---

3http://crscardellino.me/SBWCE/

---
impact it has when building neural networks. The obtained results suggest that increasing the annotated training corpus neural networks might improve the baseline models.

4. Conclusions

The main goal of this work was to develop an automatic emotion detection system from speech and language. The system acted over acoustic fragments extracted from a TV show and their corresponding transcriptions. Each fragment was annotated by means of a crowdsourcing platform using a 3-dimensional VAD model. Different neural networks architectures were tested and the obtained results show that RNN can outperform baseline systems when considering emotion detection from speech. Moreover, using a simple feedforward neural network with a very small training corpus (84 sentences) similar results to those obtained with baseline models can be achieved.

For further work we propose to get a bigger annotated corpus by using crowdsourcing tools to better train the proposed neural networks. Additionally, the two knowledge sources (acoustic and text) might be merged to provide a more accurate emotion detection system.

5. Acknowledgements

This work has been partially founded by the Spanish Government (TIN2014-54288-C4-4-R and TIN2017-85854-C4-3-R), and by the European Commission H2020 SC1-PM15 program under RIA 7 grant 69872.

6. References


Experimental Framework Design for Sign Language Automatic Recognition

Darío Tilves Santiago,1 Ian Benderitter,2 Carmen García Mateo1

1AtlanTTic Research Center – University of Vigo
2Polytech Nantes
dtilves@gts.uvigo.es, ian.benderitter@etu.univ-nantes.fr, carmen.garcia@uvigo.es

Abstract

Automatic sign language recognition (ASLR) is quite a complex task, not only for the intrinsic difficulty of automatic video information retrieval, but also because almost every sign language (SL) can be considered as an under-resourced language when it comes to language technology. Spanish sign language (SSL) is one of those under-resourced languages. Developing technology for SSL implies a number of technical challenges that must be tackled down in a structured and sequential manner.

In this paper, the problem of how to design an experimental framework for machine-learning-based ASLR is addressed. In our review of existing datasets, our main conclusion is that there is a need for high-quality data. We therefore propose some guidelines on how to conduct the acquisition and annotation of an SSL dataset. These guidelines were developed after conducting some preliminary ASLR experiments with small and limited subsets of existing datasets.

Index Terms: hearing impaired, dataset recommendations, automatic recognition, convolutional neural networks, ASL

1. Introduction

Sign language (SL) is one of the preferred ways of communication for hearing-impaired people and their families; indeed, often it is the only communication mechanism.

Automatic language recognition is highly developed for written and oral languages, but not for SLs [1]. In recent years, with the emerging interest in both neural networks (NNs) and graphical processing units (GPUs) and ongoing improvements in their efficiency, there has been significant advances in oral and written speech recognition. However, this is not the case for automatic SL recognition (ASLR). This is due to differences in recognition problems, and also to the lack of available images and videos tailored to the development of ASLR systems using machine-learning techniques.

Therefore, an important drawback to be solved is the lack, both in quantity and quality terms, of SL data for processing by deep learning algorithms. It is possible to find SL datasets, but their purpose is to be used as learning material for SL courses, rather than for automatic image recognition. The main drawback is that there are not enough repetitions of each gesture performed by one signer, and more signers are needed in order to ensure variability.

Many researchers have to find creative solutions to this problem. Some apply data augmentation to increase the volume of their data. Data augmentation [2] transforms the data in case of images by enlarging them, trimming them, changing the angle, altering the background and changing the illumination. Other possibilities are to artificially create hand gestures [3] or to record hand gestures with a device like Leap Motion which instead of acquiring images, captures the 3 dimensional movement of the hands and each finger [4], [5].

The study of SL is also constrained by its complexity. While there has been some improvement with isolated gesture recognition despite the dataset situation, there has been no advance whatsoever in continuous gesture recognition. The reason is twofold: datasets are available for isolated but not for continuous gestures, and the technology is not advanced enough to identify the latter. Nevertheless, there have been some discussions about how to proceed in continuous gesture recognition as described in [6], [7], [8] and [9].

In this paper, several datasets are described in terms of their characteristics in Section 2. Section 3 reports details of an initial study of the American SL (ASL) alphabet using transfer learning and provides a set of recommendations on how to create a robust dataset. Finally, some conclusions are drawn in Section 4.

2. Overview of existing datasets

Table 1 lists hand sign datasets (#1-#19) found in the literature, along with relevant information for its use in ASLR. The purposes of these datasets are different: some were designed for automatic recognition problems and others were created for pedagogical reasons. Shown are the dataset name and lexicon, along with the SL if any. Also provided is information on size and on variety in terms of number of signers, of repetitions per signer and type of data, with indications as to whether the dataset is labelled or not. The following will explain how Table 1 is organized.

Items #1-4 are datasets which were used in our research as training and testing material for preliminary experiments. The second column indicates the language of each dataset: items #1-11 are ASL, #12 and #13 are German SL (GSL), #14 and #15 are Spanish SL (SSL), #16 is Argentinian SL (ArsL), and #17-19 are hand gestures which do not belong to any SL. The datasets are sorted by their lexicon: from alphabets and numbers to words and sentences. All the datasets are publicly available, except for Grades Online (#14) which requires contact with the creators [10]. All the datasets are monolingual, except for SpreadTheSign (#15), which collects suggestions for signs from different SLs around the world. Some of the datasets, due to their size and/or insufficient labelling, are not completely characterized in Table 1 (indicated NA for ‘not available’).
### 3. ASLR: preliminary experiments

Learning a language is a step by step process: we first learn the alphabet and numbers, then basic words (objects, adjectives, etc.), then how to relate words (subjects, verbs, adverbs, etc.) and finally how to contextualize them, and finally correct grammar and syntax, eventually managing to make coherent conversation.

That same learning process applies to automatic speech recognition systems, in stepping up the interaction versatility with machines. And that same process ought to be applied to SL recognition systems.

We trained different NNs using some of the listed datasets in order to compare their performance. The lexicon used to train the NNs was composed of 33 ASL isolated signs: the alphabet 24 gestures and digits from 1-9. Isolated gestures were selected because they were easier to identify, as commented above. We used only one hand image as input feature to identify the isolated sign.

This experiment helped devise a proposed set of guidelines on creating a robust dataset. Also demonstrated was the usefulness of combining both depth and RGB images in datasets in our training with both types of pictures.

#### 3.1. NN architecture

An NN was first selected for the experiment. We used the Convolutional Neural Network (CNN), a deep learning approach to object identification. A CNN is based on relating a set of features to an object definition. It is designed as a set of layers containing neurons that extract features, from specific to general as the data goes through the layers. In training the CNN, this process is repeated, with the importance given to each feature changing until the CNN is capable of recognizing all the different objects provided as inputs.

We used transfer learning to demonstrate the efficiency of using trained CNNs to recognize hand signs. Transfer learning takes advantage of NNs trained for specific applications, retraining them for another application, meaning that the features extracted for the first problem are used for the new one. This is very useful to reduce the time consumed and the images needed in training because there is no need to learn all the features from scratch.

It was selected the VGG16 CNN pre-trained on the ImageNet Database [28]. Its last layer was modified in order to serve as a classifier of 33-sign ASLR experiment. The input features consist of 224x224-pixel image of one hand. So depending on the dataset, hand segmentation and resizing have to be performed.

#### 3.2. Data selection

In order to experiment with recording and segmentation techniques, an in-house dataset was acquired. This dataset is denoted UVIGO in further explanations and results. It consists of recordings of the 33 ASL signs of the lexicon described in Section 3. Kinect2 sensor was used acquiring both RGB and depth streams of the upper body. It was recorded by two signers over four days to ensure variable in recording environments (clothing, lighting conditions, etc.). 100 repetitions of each sign were used for training and 10 repetitions for testing.

Of the available datasets described in Section 2, two were selected for NN training and testing: Superpixel (#1), composed of alphabetical RGB images of one hand, and Fingerspelling (#2), containing both alphabetical and numerical depth pictures.

---

Table 1: Summary of datasets found in the literature.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Language and Lexicon</th>
<th>Size</th>
<th>Data Type</th>
<th># Repetitions</th>
<th># Signers</th>
<th>Labelled</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Superpixel [11]</td>
<td>ASL: 24 alphabet signs</td>
<td>131688 images</td>
<td>RGB + Depth</td>
<td>500</td>
<td>5</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Fingerspelling [12]</td>
<td>ASL: 24 alphabet signs and digits 1-9</td>
<td>31000 images</td>
<td>Depth</td>
<td>200</td>
<td>5</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Massey University [13]</td>
<td>ASL: 24 alphabet signs and numbers 0-9</td>
<td>2524 images</td>
<td>RGB</td>
<td>5</td>
<td>5</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Padova Senz3D [3]</td>
<td>ASL: letters B, D, I, S, and digits 2, 3, 4, 5, 9, 10</td>
<td>2640 images</td>
<td>RGB + Depth</td>
<td>30 + 30</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Padova Kinect [14]</td>
<td>ASL: letters A, D, I, L, W, Y, and digits 2, 5, 7</td>
<td>2800 images</td>
<td>RGB + Depth</td>
<td>10 + 10</td>
<td>14</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>HKU Kinect Gesture [15]</td>
<td>ASL: letters A, L, Y and digits 1-5</td>
<td>3000 images</td>
<td>RGB</td>
<td>60</td>
<td>5</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>NTU Microsoft Kinect [16]</td>
<td>ASL: letters A, L, Y and digits 1-5</td>
<td>2000 images</td>
<td>RGB + Depth</td>
<td>10 + 10</td>
<td>10</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>Cypr15 [17]</td>
<td>ASL: 7 words and digits 1-9</td>
<td>68000 images</td>
<td>Depth</td>
<td>500</td>
<td>8</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>RWTH-50 [18]</td>
<td>ASL: 83 words</td>
<td>8844 videos</td>
<td>Grey scale</td>
<td>2</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>10</td>
<td>ASLLVD [19]</td>
<td>ASL: words</td>
<td>992 videos</td>
<td>RGB</td>
<td>2</td>
<td>5</td>
<td>Yes</td>
</tr>
<tr>
<td>11</td>
<td>RWTH-104 [20]</td>
<td>ASL: 201 sentences</td>
<td>201 videos</td>
<td>Grey scale</td>
<td>1</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>12</td>
<td>German Spelling [21]</td>
<td>GSL: 30 alphabet signs and digits 1-5</td>
<td>3080 videos</td>
<td>Grey scale</td>
<td>2</td>
<td>20</td>
<td>Yes</td>
</tr>
<tr>
<td>13</td>
<td>RWTH PHOENIX [22]</td>
<td>GSL: sentences</td>
<td>592383 images</td>
<td>RGB</td>
<td>1</td>
<td>NA</td>
<td>No</td>
</tr>
<tr>
<td>14</td>
<td>Grades Online [10]</td>
<td>SSL: 750 words</td>
<td>750 videos</td>
<td>RGB</td>
<td>1</td>
<td>2</td>
<td>Yes</td>
</tr>
<tr>
<td>15</td>
<td>SpreadTheSign [23]</td>
<td>SSL: words and sentences</td>
<td>+2000 videos</td>
<td>RGB</td>
<td>1</td>
<td>NA</td>
<td>Yes</td>
</tr>
<tr>
<td>16</td>
<td>LSAG6 [24]</td>
<td>ArSL: 64 words</td>
<td>3200 videos</td>
<td>RGB</td>
<td>5</td>
<td>10</td>
<td>Yes</td>
</tr>
<tr>
<td>17</td>
<td>SKIG [25]</td>
<td>Hand gestures: 10</td>
<td>2160 videos</td>
<td>RGB + Depth</td>
<td>18 + 18</td>
<td>6</td>
<td>Yes</td>
</tr>
<tr>
<td>18</td>
<td>Microsoft Gesture-RC [26]</td>
<td>Hand gestures: 12</td>
<td>594 videos</td>
<td>RGB + Depth</td>
<td>NA</td>
<td>NA</td>
<td>No</td>
</tr>
<tr>
<td>19</td>
<td>Chatearn 2016 [27]</td>
<td>Hand gestures</td>
<td>140945 videos</td>
<td>RGB + Depth</td>
<td>1 + 1</td>
<td>NA</td>
<td>No</td>
</tr>
</tbody>
</table>
of one hand. 100 randomly-selected repetitions of each sign were used for training and 10 for testing.

Two further datasets were selected only as test datasets: the RGB images of Massey University dataset (#3), and the depth images of Padova Senz3D dataset (#4).

3.3. Data preprocessing

Image preprocessing in machine learning typically consists of cropping, resizing and feature extraction operations. Next we describe the preprocessing tasks needed for each dataset.

As said before, the used CNN required 224x224-pixel images as input features, so resizing had to be performed for Superpixel, Fingerspelling and Massey University datasets. Cropping (hand segmentation) is not needed for these datasets.

Regarding UVIGO dataset, hand segmentation is made using the information provided by the Kinect2 sensors. Kinect2 gives access to RGB and depth image streams, along with 25 body joint coordinates. In the RGB stream our segmentation algorithm uses 4 body joints to easily locate a hand. Then it crops a square around the detected hand. For the depth stream we use again 4 joints to locate the hand. This hand is segmented in the xy plane as the RGB images. The algorithm applies another cropping in the z axis using the depth values of the hand to set a threshold (Figure 1). This eliminates the background from the depth images.

Finally, we also performed a segmentation in Padova dataset. It was applied the depth stream cropping described before, but in this case instead of using the joints provided by Kinect2, we directly used the depth values to distinguish the hand from its surroundings.

![Segmentation representation.](image)

3.4. Experiments and results

The results presented in this section are accuracy scores calculated from CNN predictions. All the CNNs were trained in the same way, fixing the number of epochs to five, and with a low initial learning rate, which is divided by four every epoch. With such a reduced number of epochs, and taking advantage of the higher learning slope that characterize transfer learning, the goal was to reduce overfitting.

In a first experiment, each training dataset was separately used to train a CNN. Table 2 shows the results in terms of accuracy over the testing data.

The highest scores in Table 2 are from testing each CNN with its testing material (main diagonal). However, despite UVIGO CNN having the lower score of these three, it performed better when testing it with the #1 and #2 datasets (second row in Table 2). Counterwise, Fingerspelling CNN had the highest accuracy (99.61%) but the lowest accuracy values over the other datasets (fourth row in Table 2).

UVIGO CNN (which was trained using both RGB and depth images) tested over the other datasets gave better results, which it seems that generalizes better. Therefore, it resulted that training with RGB pictures or depth pictures separately did not perform well in recognizing depth and RGB pictures, respectively.

As seen in Table 1, #1 and #2 both have five signers unlike UVIGO which has two signers. More signers mean more variation, and therefore more generalization. However, these results seem to indicate that using RGB and depth images is as important as the number of signers.

<table>
<thead>
<tr>
<th>Training data/Testing data</th>
<th>UVIGO</th>
<th>Superpixel</th>
<th>Fingerspelling</th>
</tr>
</thead>
<tbody>
<tr>
<td>UVIGO</td>
<td>89.76%</td>
<td>27.97%</td>
<td>20.96%</td>
</tr>
<tr>
<td>Superpixel</td>
<td>17.39%</td>
<td>92.48%</td>
<td>7.81%</td>
</tr>
<tr>
<td>Fingerspelling</td>
<td>10.00%</td>
<td>6.38%</td>
<td>99.61%</td>
</tr>
</tbody>
</table>

In a second experiment, the three training datasets were jointly used to train three CNNs.

Table 3 summarizes results for the training of these three CNNs: “All RGB” label means a CNN trained with only RGB images from UVIGO and Superpixel datasets, “All depth” label means a CNN trained with only depth images from UVIGO and Fingerspelling datasets, and “All” label means a CNN trained with both types of images. They are also compare to Massey (#3) and Padova (#4) datasets.

It can be seen that the CNNs based on both RGB and depth images outperformed the CNNs trained with only one kind of image. All (Table 3) exceeded the 50% precision mark on external datasets. All depth also did so for depth, but only for the Padova dataset because it consists of depth images.

<table>
<thead>
<tr>
<th>Training data/Testing data</th>
<th>All RGB</th>
<th>All depth</th>
<th>Massey</th>
<th>Padova</th>
</tr>
</thead>
<tbody>
<tr>
<td>All RGB</td>
<td>91.62%</td>
<td>6.56%</td>
<td>25.53%</td>
<td>7.31%</td>
</tr>
<tr>
<td>All depth</td>
<td>8.53%</td>
<td>95.33%</td>
<td>16.73%</td>
<td>59.70%</td>
</tr>
<tr>
<td>All</td>
<td>95.21%</td>
<td>96.77%</td>
<td>52.51%</td>
<td>69.63%</td>
</tr>
</tbody>
</table>

It can be concluded that mixing depth and RGB images is an effective way to create a robust and flexible dataset to train systems for SL recognition. However, if only one type of image must be chosen, it is preferable to work with depth datasets because overall have better performance than RGB datasets. This may be due to the absence of background noise which was filtered while preprocessing.

3.4.1. Model application

The best trained network was incorporated in a hand sign recognition application, resulting in segmentation speeds of 0.001s for RGB images and 0.007s for depth images (both
hands at the same time), and a prediction time of 0.054s. The whole process was therefore performed in 0.116s.

3.5. Dataset acquisition guidelines

As mentioned, training a sign recognition NN requires significant amounts of data, with sufficient diversity in sign execution and in recording environments to ensure robustness. Considering the results, dataset acquisition guidelines are described next to meet these requirements.

It is therefore recommended to record data from at least 10 different signers and to ensure a balance between male and female and native and non-native signers. It is also recommended to spread recording sessions over several days to ensure a variety of conditions (lighting, clothes, etc.) and settings.

It is highly recommended to record RGB and depth at the same time, preferably with a resolution of at least 640x480 pixels. Depth images are useful because a more precise segmentation is possible. Another advantage is that different lightning conditions, clothes and settings do not have any influence because they are not perceived by the sensors. Nonetheless, recording sex, height and anatomical differences is important.

To train the system for each sign the same way, it is highly recommended to formalize the number of images recorded per sign, signer and recording setting. 100 images per isolated gesture and 20 videos per continuous gesture for each signer is proposed as a good compromise. For a sufficient number of signers, this represents a sufficiently large and sufficiently heterogeneous set of empirical data to avoid training the network with images too similar to each other.

Formalization also requires efficient and convenient organization and annotation. It is therefore recommended to do the following: organize the dataset in a folder tree, with a single folder per signer, and with each containing a sub-folder per sign and sub-sub-folders for each repetition of that sign; identify signer folders using an identification code, and name sign folders after the sign name; identify each image with the signer identification code, the recording date (or code), the repetition number and the sign name; and finally, if feasible, assign each recorded sign a code and use that instead of its name.

It is recommended describe files in terms of three sections: with the lexicon and the associated code; with the signer’s name and respective code with information on date of birth, why they use SL and their dominant hand; and with details on the recording session, equipment used, recorder name, recording date, file name and recording conditions. Recommended is to use a spreadsheet (or CSV) because it is very easy to filter out and retrieve key statistics from the data.

4. Conclusions and further work

Although a few studies exist on ASLR, progress is still limited by the lack of datasets. The fact that the few available lack reliability and convenience makes the creation of new datasets mandatory in order to progress in this field.

We have described an experimental framework designed to study the reliability of existing datasets and the combination of RGB and depth data and experimentally trained CNNs. The system achieved over 50% precision for challenging datasets from both natural and artificial recording environments. This method can be used to create a complete and straightforward dataset suitable for research.

Future research will focus on improving system training to match or surpass accuracy rates for both depth and RGB images separately with all datasets included and on extending recognition to the Spanish SL. An expansion of the acquired dataset is also planned, as well as the creation of two computer applications, one for recording and another for real-time gesture recognition.

5. Acknowledgements

This work has received financial support from the Xunta de Galicia (Agrupación Estratéxica Consolidada de Galicia accreditation 2016-2019, Galician Research Network TecAnDaLi ED431D 2016/011 and Grupos de Referencia Competitiva GRC2014/024) and the European Union (FEDER/ERDF).

6. References


Baseline Acoustic Models for Brazilian Portuguese Using Kaldi Tools

Cassio T. Batista, Ana Larissa da S. Dias, Nelson C. Sampaio Neto

Federal University of Pará, Institute of Exact and Natural Sciences
Augusto Corrêa 1, Belém 660750-110, Brazil
{cassiotb,nelsonneto}@ufpa.br, ana.dias@itec.ufpa.br

Abstract

Kaldi has become a very popular toolkit for automatic speech recognition, showing considerable improvements through the combination of hidden Markov models (HMM) and deep neural networks (DNN). However, in spite of its great performance for some languages (e.g., English, Italian, Serbian, etc.), the resources for Brazilian Portuguese (BP) are still quite limited. This work describes what appears to be the first attempt to create Kaldi-based scripts and baseline acoustic models for BP using Kaldi tools. Experiments were carried out for dictation tasks and a comparison to CMU Sphinx toolkit in terms of word error rate (WER) was performed. Results seem promising, since Kaldi achieved the absolute lowest WER of 4.75% with HMM-DNN and outperformed CMU Sphinx even when using Gaussian mixture models only.

Index Terms: automatic speech recognition, Brazilian Portuguese, Kaldi

1. Introduction

The attempts to simplify the communication between humans and machines are not a novelty. Ever since the emergence of consumer electronic computers, researchers have been doing a lot of effort in order to design more convenient interfaces for controlling and interacting with electronic devices. However, despite the consolidation of keyboards and mice as main input methods for personal computers, as well as touch screens for mobile devices, alternative control interfaces such as speech, body gestures, or even thoughts never ceased to be investigated.

For years, the combination of hidden Markov models (HMM) and Gaussian mixture models (GMM) has been the state-of-the-art technique for acoustic modeling in the automatic speech recognition (ASR) field [1, 2]. Concerning the Brazilian Portuguese language in particular, robust speech recognition systems based on traditional HMM-GMMs have already been proposed [3, 4] using both HTK [5] and CMU Sphinx [6] tools. However, the deep learning approaches that emerged last decade seem to have outperformed HMM-GMM models when replacing the Gaussian mixtures by deep neural networks (DNN) combined with HMMs.

Most works that apply deep learning for speech recognition make use of Kaldi [7], an open-source software package that implements the hybrid HMM-DNN combination. To the best of our knowledge, no previous work has developed a Kaldi recipe for Brazilian Portuguese (BP) yet. Therefore, towards building freely available resources [8] for a large vocabulary continuous speech recognition (LVCSR) system in BP using the Kaldi toolkit, this work presents results for HMM-GMM triphone-based acoustic models in terms of word error rate (WER). A preliminary result for DNN-based models is also presented, but the main development of HMM-DNN hybrid models is still ongoing, given the huge amount of time taken to train them.

2. Related Work

A literature review was conducted on IEEE Xplore and ACM Digital Library. However, most works from ACM simply mention speech recognition as an application of DNNs. Therefore, only papers from IEEE Xplore were considered. After filtering by title and abstract, the most relevant ones were selected and will be shortly detailed below.

Sahu and Ganesh [9] performed a survey on HTK, CMU Sphinx and Kaldi toolkits for different languages regarding their performance in terms of WER. They found that Kaldi achieved the best WER value of 2.7% using the Wall Street Journal (WSJ) English corpus. In another work, Becerra et al. [10] presented a comparative case study for Spanish between the conventional HMM-GMM architecture and the recent HMM-DNN model using Kaldi. The audio corpus used includes 1,836 sentences from 87 speakers sampled at 16 kHz, which are a mixture of human voices and text-to-speech utterances. A 20.71% improvement was achieved by the HMM-DNN architecture over the HMM-GMM models: 3.33% against 4.20% of WER, respectively.

Popović et al. [11] used Kaldi to develop an HMM-based ASR system for the Serbian language. The audio corpora used in the experiment contains 95 hours of speech sampled at 8 kHz. They obtained a word recognition accuracy of approximately 98%. For Italian, on the other hand, Cosi [12] adapted Kaldi’s TIMIT recipe for the FBK ChildIt corpus, which contains approximately 10 hours of speech of children sampled at 16 kHz. The results only show that DNN configurations outperform the non-DNN ones. Karan et al. [13] also used Kaldi to develop a speech recognition system, now for Hindi Odia language. The audio corpus consisted of 2,647 utterances collected from 104 speakers at 8 kHz using mobile phones. The experiment used the conventional HMM-GMM architecture only, and reports the best result of 1.74% WER in the triphone model.

Ali et al. [14] presented a complete Kaldi recipe for building Arabic speech recognition systems. The corpus used was the GALE Arabic Broadcast News data set, which consisted of 100,000 speech segments of nine different TV channels, a total of 203 hours of speech data recorded at 16 kHz. In the experiment, the DNN-based system achieved the best results with an overall WER of 26.95%, which is nearly a 10% relative improvement to the HMM-GMM model. Kipyatkova and Karpov [15], on the other hand, built an HMM-DNN acoustic model using Kaldi for Russian language. For training and testing, they used their own speech recorded at 44.1 kHz with 16 bits per sample. The data set was composed by 55 speakers and 16,850 utterances. Two different kinds of neuron activation functions were implemented on the neural network: tanh and p-norm. The results showed that the p-norm function obtained the best WER value of 20.30%.

Another search was performed on the previous IberSPEECH proceedings of 2014 and 2016, where two works stood out. On the first one, Guiroy et al. [16] im-

77 10.21437/IberSPEECH.2018-17
implemented an HMM-DNN ASR system in Kaldi and also conducted a comparative study between HMM-based models in both Kaldi and HTK. The Castillian Spanish SpeechDat(II) FDB-4000 audio corpus was used, which contains 43 hours of recordings from 4,000 speakers. The results indicated a 34.02% decrease in WER when comparing the most accurate DNN-based and HMM-based models from Kaldi. A decrease of 53.79% for the HMM-based model in Kaldi could also be observed over their most accurate model from HTK.

On the second work, Zorrilla et al. [17] carried out several experiments using Kaldi in order to evaluate different deep-learning approaches for acoustic modeling on well-known Spanish data sets, namely Albayzin, Dihana, CORLEC-EHU and TC-STAR. In addition, the El País text corpus was used for language modeling. The authors found through experiments that all HMM-DNN hybrid acoustic models have outperformed the HMM-GMM ones and work well even with non-task-specific language models.

During the research, we also found two works that tackle the ASR problem for BP using deep neural networks. Quintanilha et al. [18] presented an open-source, character-based, end-to-end bidirectional long short-term memory (BLSTM) neural network for LVCSR. Several experiments were conducted over a data set of approximately 14 hours of recorded audio and the best performance evaluated in terms of label error rate was 31.53% without the use of any language model. Bonilla et al. [19], on the other hand, proposed an end-to-end deep-learning system for recognizing digits, which is compared to a simple multilayer perceptron (MLP) network. It is not clear, however, if the system classifies characters, words or phonemes. The best result is reported as 97.5% of accuracy rate, against 82.8% achieved by the MLP.

According to the literature review, it appears no previous work has developed ASR resources with Kaldi for Brazilian Portuguese yet. Therefore, we believe this is the first attempt to build acoustic models for BP using the toolkit’s deep learning approaches.

3. Tools and Resources for BP using Kaldi

In order to build a speech recognition system, one must be provided with a language model (LM), a phonetic dictionary and an acoustic model (AM). The resources and tools used to build each one of the three aforementioned components with Kaldi will be detailed below. It is worth mentioning that the LM and the dictionary are the very same used in CMU Sphinx as well. The steps to train the AMs in particular are similar for both toolkits, but some differences will be pointed out along the text. For further information about acoustic model training for BP using CMU Sphinx tools, the reader is referred to [4].

3.1. Audio Corpora

Speech recognition is a data-driven technology, which means it requires a relatively large amount of labeled data (transcribed audio) to work properly. The corpora used to train the acoustic models with Kaldi are composed by seven data sets, as summarized in Table 1. The data sets contain audio files in an uncompressed, linear, signed PCM (namely, WAV) format, and are sampled at 16 kHz with 16 bits per sample. It is important to note that the actual number of speakers in West Point was rather reduced due to abundance of foreign words amidst the corpus. Besides, Constitution and Consumer Protection Code corpora share the same speaker.

3.2. Phonetic Dictionary and Language Model

The phonetic dictionary maps every grapheme in the lexicon (orthographic representation) to one or more phonetic transcriptions. The software described in [24] was used to include the pronunciation mapping of each of the 14,518 words into the dictionary. The trigram language model used in this work is described in [3]. It was trained with the SRILM [25] toolkit with 1.6 million phrases from the CETENFolha [26] corpus, yielding a perplexity value of 170. The LM is available in ARPA format, but in order to be used on the Kaldi environment, it was converted to the FST format using the provided arpa2fst script.
Table 1: Audio corpora used to train acoustic models.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Ref.</th>
<th>Hours</th>
<th>Words</th>
<th>Speakers</th>
</tr>
</thead>
<tbody>
<tr>
<td>LapsStory</td>
<td>[3]</td>
<td>5h:18m</td>
<td>8,257</td>
<td>5</td>
</tr>
<tr>
<td>LapsBenchmark</td>
<td>[3]</td>
<td>0h:54m</td>
<td>2,731</td>
<td>35</td>
</tr>
<tr>
<td>Constitution</td>
<td>[20]</td>
<td>8h:58m</td>
<td>5,330</td>
<td>1</td>
</tr>
<tr>
<td>Consumer</td>
<td>[20]</td>
<td>1h:25m</td>
<td>2,003</td>
<td>1</td>
</tr>
<tr>
<td>Protection Code</td>
<td>[20]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spoltech LDC</td>
<td>[21]</td>
<td>4h:19m</td>
<td>1,145</td>
<td>475</td>
</tr>
<tr>
<td>West Point LDC</td>
<td>[22]</td>
<td>5h:22m</td>
<td>484</td>
<td>70</td>
</tr>
<tr>
<td>CETUC</td>
<td>[23]</td>
<td>14h:39m</td>
<td>3,528</td>
<td>101</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>170h:51m</td>
<td>14,518</td>
<td>687</td>
</tr>
</tbody>
</table>

3.3. Acoustic Model

The scripts used on the Brazilian Portuguese audio corpora were based on Kaldi’s recipe available for the WSJ corpus [27]. For the sake of comparison, HMM-GMM acoustic models were trained with both Kaldi and CMU Sphinx as well. On Kaldi, the deep learning approach actually uses the HMM-GMM training as a pre-processing stage. Figure 1 shows the steps followed to train HMM-DNN acoustic models on Kaldi based on HMM-GMM triphones. The audio signals are windowed at every 25 ms with 10 ms of overlap in the front-end, being encoded as a 39-dimension vector: 12 Mel frequency cepstral coefficients (MFCCs) [28] using C0 as the energy component, plus 13 delta (Δ, first derivative) and 13 acceleration (ΔΔ, second derivative) coefficients are extracted from each window.

The AMs are iteratively refined. The flat-start approach models 39 phonemes (38 monophones plus one silence model) as context-independent HMMs. The standard 3-state left-to-right HMM topology with self-loops was used. At the flat-start, a single Gaussian mixture models each individual HMM with the global mean and variance of the entire training data. The transition matrices are also initialized with equal probabilities. The parameters used for extracting MFCCs and training the monophones are the same for both CMU Sphinx and Kaldi. Nevertheless, Kaldi uses the Viterbi training algorithm [29] to re-estimate the models at each training step, rather than the Baum-Welch algorithm [30] used by CMU Sphinx. Furthermore, Viterbi alignment is applied after each training step in order to allow training algorithms to improve the model parameters, a feature that is not present on CMU Sphinx by default. Subsequently, the context-dependent HMMs are trained for each triphone, first with the delta and after with the acceleration coefficients. Each triphone is represented by a leaf on a decision tree, which is automatically created by both toolkits using statistical methods. Eventually, leaves with similar phonetic characteristics are then tied/clustered together.

The last two steps for training a HMM-GMM acoustic model with Kaldi are the linear discriminant analysis (LDA) [31] combined with the maximum likelihood linear transform (MLLT) [32], followed by the speaker adaptive training (SAT) [33]. Both are included in most tutorials for AM training with Kaldi. The latter, however, was not taken into account during our simulations in order to save time, so only LDA+MLLT was adopted. Moreover, these two steps are not enabled by default on CMU Sphinx and, since they were not used in [4] either, we decided not to include them in order to try to reproduce the results and to save time as well.

Table 2: Kaldi DNN tools and parameters used for training.

<table>
<thead>
<tr>
<th>Tool or Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN codebase</td>
<td>nnet2 (&quot;Dan’s DNN&quot;)</td>
</tr>
<tr>
<td>Script</td>
<td>train_pnorm_fast.sh</td>
</tr>
<tr>
<td>Hidden layers</td>
<td>2</td>
</tr>
<tr>
<td>Activation function</td>
<td>p_norm</td>
</tr>
<tr>
<td>pnorm_output_dim</td>
<td>3,000</td>
</tr>
<tr>
<td>pnorm_input_dim</td>
<td>300</td>
</tr>
<tr>
<td>num_epochs</td>
<td>8</td>
</tr>
<tr>
<td>num_epochs_extra</td>
<td>5</td>
</tr>
<tr>
<td>Minibatch size</td>
<td>512</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.02 down to 0.004</td>
</tr>
</tbody>
</table>

The LDA technique takes the feature vectors and splice them across several frames, building HMM states with a reduced feature space. Then, a unique transformation for each speaker is obtained by a diagonalizing MLLT transform. On top of the LDA+MLLT features, the fMLLR alignment algorithm, which is a speaker normalization that uses feature-space maximum likelihood linear regression (MLLR), is applied [34].

Finally, the HMM-DNN acoustic model is obtained by using the neural network to model the state likelihood distributions as well as to input those likelihoods into the decision tree leaf nodes [16]. In short terms, the network input are groups of feature vectors and the output is given by the aligned state of the HMM-GMM system for the respective features of the input. The number of HMM states in the system also defines the DNN’s output dimension [15].

Table 2 shows the most important Kaldi tools and parameters set used to train the deep neural network. Kaldi provides two distinct implementations for DNN training: nnet1 [35], which is primarily maintained by Katel Veselý; and nnet2 [36], by Daniel Povey. The former was chosen because it supports CPU training while the nnet1 enables GPU training only, a resource that was not available for us. Regarding the activation functions of the DNN, we chose the p_norm nonlinearity because it presents a superior performance over the tanh in the literature review [15, 16, 17]. The remaining parameters of the DNN were set based in the Kaldi’s documentation as well as in the related works. Since there is actually no parameter to define the number of neurons in the hidden layers for p-norm networks, pnorm_output_dim and pnorm_input_dim parameters must be set instead, being the latter an integer multiple of the former usually with a ratio of 5 or 10 [37]. The number of epochs is given by the sum of the num_epochs and num_epochs_extra parameters. The first one was supposed to be 15, but it is recommended to reduce it when the computational environment is not very high powered [37], so we choose 13 (8+5) to be the total number of epochs. The learning rate was set to vary from 0.02 down to 0.004 during the default number of epochs; and to stay constant at 0.004 for the next extra epochs [15].

4. Experimental Tests and Results

Tests were executed on an HP EliteDesk 800 G1 desktop computer equipped with an Intel® Core™ i5-4570 3.20 GHz CPU, 8 GB of RAM and 1 TB of hard disk storage. During the experiments, the LapsBenchmark corpus was held exclusively for
testing and the six other corpora were used for training. Unfortunately, no clusters or graphic cards could be used for training the models. Therefore, due to the computational burden and the lack of hardware resources, it was not possible to develop DNN-based AMs for all combinations of HMM-GMM acoustic models with Kaldi.

Table 3 shows the results obtained with both CMU Sphinx and Kaldi. For Kaldi, by the way, the WER was evaluated across all triphone training steps in order to perform a more complete comparison to CMU Sphinx results, since neither the LDA+MLLT stage or the fMLLR alignment were included for this toolkit. For Sphinx, as expected, the WER decreases as we increase both the number of Gaussians and the number of tied-states of the model. However, the values seem to converge after 4,000 senones and 8 Gaussians. The lowest WER value achieved was approximately 11.1% with 4,000 senones and 16 Gaussian densities.

For Kaldi, however, we found that the previous convergence shown on CMU Sphinx results does not occur. As we increase the number of senones and the number of Gaussians, the WER values linearly drop. Besides, it can be seen that the lowest WER values for the first two triphone training steps (tri-Δ and tri-ΔΔ) are already lower than the best one achieved by CMU Sphinx: 9.31% and 9.23%, respectively. The global, lowest WER value obtained with Kaldi was 6.5% with 8,000 tied-states and 16 Gaussians at the tri-LDA+MLLT step, which is equivalent to 128,000 leaves on the decision tree, according to Kaldi’s parameter settings (which is basically the result of the product 8,000 × 16).

As proof of concept, we trained a DNN-based acoustic model on the best HMM-GMM model produced with Kaldi. The WER value dropped from 6.5% to 4.75%, an improvement of 26.92%. When compared to the lowest WER value obtained with CMU Sphinx, the improvement increases to 57.21%, which is a huge difference for dictation tasks.

5. Conclusions and Future Works

This paper addressed the first attempt to develop a speech recognition system for large vocabulary (LVCSR) in Brazilian Portuguese using the Kaldi toolkit. Triphone-based, HMM-GMM acoustic models with different values of Gaussians and tied-states were trained with Kaldi and CMU Sphinx tools in order to establish a comparison in terms of word error rate (WER). The evaluation results showed that the systems perform better as we increase the number of Gaussian densities per mixture and the number of tied-states. For CMU Sphinx, the results obtained are in accordance to [4], in spite of the current WER achieved being lower, possibly due the larger corpora used for training the models.

Results also showed that Kaldi definitely outperformed CMU Sphinx even without the use of its deep learning tools. An explanation might be the use of Viterbi algorithm for training (rather than Baum-Welch), as well as the use of Viterbi alignments in between each training stage, which is said to improve or refine the parameters of the model [38]. With the use of DNNs, Kaldi presents an improvement of 57.21% over the best HMM-GMM-based acoustic model built with CMU Sphinx.

As future work, we plan to finish training the HMM-DNN triphone-based AMs with Kaldi and consequently make them publicly available (together with the recipe) [8] to the community. We also expect to test with 32 and 64 densities per mixture, now evaluating the decoding time too in terms of the real-time factor (xRT) as the WER possibly decreases. Furthermore HTK’s latest release also has an implementation of deep learning algorithms, which may join the next comparisons.

6. Acknowledgements

The authors would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Federal University of Pará (UFPA) under Edital nº 06/2017 – PIBIC/PROPESP for granting scholarships.
7. References


Converted Mel-Cepstral Coefficients for Gender Variability Reduction in Query-by-Example Spoken Document Retrieval

Paula Lopez-Otero¹, Laura Docio-Fernandez²

¹Universidade da Coruña - CITIC, Information Retrieval Lab
²Universidade de Vigo - atlanTTic Research Center, Multimedia Technology Group

paula.lopez.otero@udc.gal, ldocio@gts.uvigo.es

Abstract

Query-by-example spoken document retrieval (QbESDR) is a task that consists in retrieving those documents where a given spoken query appears. Spoken documents and queries exhibit a huge variability in terms of speaker, gender, accent or recording channel, among others. According to previous work, reducing this variability when following zero-resource QbESDR approaches, where acoustic features are used to represent the documents and queries, leads to improved performance. This work aims at reducing gender variability using voice conversion (VC) techniques. Specifically, a target gender is selected, and those documents and queries spoken by speakers of the opposite gender are converted in order to make them sound like the target gender. VC includes a resynthesis stage that can cause distortions in the resulting speech so, in order to avoid this, the use of the converted Mel-cepstral coefficients obtained from the VC system is proposed for QbESDR instead of extracting acoustic features from the converted utterances. Experiments were run on a QbESDR dataset in Basque language, and the results showed that the proposed gender variability reduction technique led to a relative improvement by 17% with respect to using the original recordings.

Index Terms: query-by-example spoken document retrieval, dynamic time warping, voice conversion, variability compensation

1. Introduction

Spoken document retrieval (SDR) consists in, given a set of spoken documents, retrieving those documents where a given query appears. The availability of technologies to perform this task is of paramount importance nowadays due to the amount of multimedia contents that are part of our everyday life. STD can be carried out using either written or spoken queries. The latter alternative, known as query-by-example SDR (QbESDR), allows a natural communication with devices while easing the access to such technologies to visually impaired users. For these reasons, this task has gained the attention of the research community, which led to the organization of evaluations in order to encourage investigation on this topic [1, 2, 3, 4, 5, 6, 7].

The most common strategies for QbESDR are based on pattern matching techniques: first a set of features is extracted from the documents and queries, and then each query-document pair is compared using an alignment algorithm usually based on dynamic time warping (DTW) [8] or any of its variants. These techniques usually rely on posteriorgram representations of the speech utterances such as phone posteriorgrams [9], which account for the probability of each phone unit in a phone decoder given a speech frame [9, 10, 11]; and Gaussian posteriorgrams, which represent the likelihood of each Gaussian in a Gaussian mixture model (GMM) given a speech frame [12, 13, 14, 15].

Gaussian posteriorgrams are computed from acoustic features extracted from the waveforms, so they can be considered a zero-resource representation for QbESDR: this is interesting since no linguistic resources are necessary to develop the system. However, the use of acoustic features for speech representation makes the system sensitive to different sources of variability such as speaker identity, gender, accent or recording channel [16].

Previous work showed that compensating the query and document variability in terms of gender leads to an improvement of QbESDR performance [16]. In those experiments, alternative queries were generated via voice conversion (VC) in order to have a female and male version of every query; then, male queries were searched within male documents and female queries within female documents. These results led to believe that transforming both queries and documents into a similar voice would reduce the acoustic variability even more. Some experiments following this research direction were presented in [14], where vocal tract length normalization (VTLN) was used to reduce the speaker-specific variation in the queries and documents: this procedure implies computing the warping factor of every recording. In this paper, a new gender variability compensation strategy is proposed that consists in, given a target gender, converting all the documents and queries of the opposite gender to that target gender so that all the resulting recordings will all be spoken by speakers of the same gender. For this purpose, a speaker-independent VC strategy [17] is used, which allows the use of the same conversion function for all the spoken utterances.

Typical VC techniques consist in a speech analysis stage followed by feature conversion and speech resynthesis in order to obtain new speech utterances with the converted speech. One of the main drawbacks of these strategies is the negative effect caused by the vocoder since it can introduce distortions that reduce the quality and intelligibility of the resulting spoken utterances [18]. Nevertheless, it would not be necessary to resynthesize the speech if the features used in the VC procedure are suitable for speech representation in the QbESDR task. In this situation, the use of Gaussian posteriorgrams for speech representation is straightforward, since it allows the modelling of the documents and queries using the converted features obtained from the VC system. Therefore, in this paper, a comparison of QbESDR results when using converted features and when extracting the same features from the converted waveforms is made in order to quantify the influence of the speech resynthesis stage.

The rest of this paper is organized as follows: Section 2 describes the proposed gender variability compensation strategy for QbESDR; the experimental framework used for validation is described in Section 3; Section 4 discusses the experimental results; and, finally, some conclusions and future work are...
presented in Section 5.

2. Gender variability compensation in QbESDR

The gender variability compensation technique proposed in this paper is depicted in Figure 1: given a target gender, all the speech utterances that are spoken by speakers of the opposite gender are converted to the target gender via VC, while the others remain unchanged. Then, QbESDR is performed using the resulting documents and queries. In this way, all the speech utterances sound as spoken by speakers of the same gender, therefore reducing the gender variability of the recordings in the QbESDR task.

The implementation of this system requires three different tasks: gender classification, gender conversion, and search.

2.1. Gender classification

First, the gender of the documents and queries must be detected in order to find out whether a speaker belongs to the target gender or, on the contrary, the speech must be converted. In this work, a gender classifier based on the GMM log-likelihood ratio was used. Given a speech utterance to classify, its likelihoods given male and female GMMs are computed and then the log-likelihood ratio is calculated. The utterance is classified as belonging to the gender with the highest likelihood [19].

As mentioned in [16], the proposed gender classifier has shown an accuracy close to 100%, so few utterances are expected to be misclassified. Nevertheless, since the proposed approach aims at converting the gender of the utterance, the apparent gender is more relevant than the actual gender, so possible classification errors are not really important in this work.

2.2. Gender conversion

Male and female voices are different for anatomical reasons: males usually have longer vocal tracts as well as longer and heavier vocal folds than females. This derives in differences in the fundamental frequency (F0) and formant frequencies of their voice, which are generally higher for women [20]. Hence, VC techniques can be used to transform the gender of a speaker by modifying the voice characteristics of a source speaker in order to make it sound like a target speaker. In previous work on gender conversion for QbESDR, the VC technique proposed in [17] was used since it is speaker-independent, i.e. it can be used to transform any speaker into a different (undetermined) one of the opposite gender.

The gender conversion system used in this system has three stages, as most VC strategies: speech analysis, feature conversion, and speech resynthesis, as depicted in Figure 2. Three different sets of features are commonly extracted in the first stage [21]: 40 Mel-cepstral coefficients (MCEP), fundamental frequency (F0) and band aperiodicity features (BAP).

The VC approach used in this work is based on frequency warping (FW) and amplitude scaling (AS), and it consists in applying an affine transformation in the cepstral domain [22]:

\[ y = Ax + b \]  

(1)

where \( x \) is a MCEP vector, \( A \) denotes a FW matrix, \( b \) represents an AS vector, and \( y \) is the transformed version of \( x \).

The method proposed in [17] uses a simplified FW curve that is defined piecewise by means of three linear functions, as depicted in Figure 3: the discontinuities of the FW curve are placed at frequencies \( f_a \) and \( f_b \); \( \alpha \) is the angle between the 45-degree line and the first linear function; and \( \beta \) is the angle between the 45-degree line and the second linear function, defined as \( \beta = k \alpha \), where \(|k| < 1 \). Values of \( \alpha \) greater (less) than 0 lead to higher (lower) formant frequencies, resulting in a male-to-female (female-to-male) conversion function. This strategy is similar to vocal tract length normalization but, in this case, the same parameters are used for all the speech utterances, avoiding the need to compute a warping factor for each speaker.

Afterwards, the AS vector \( b \) is defined by selecting random values from a set of weighted Hanning-like bands equally spaced in the Mel-frequency scale [23] as described in [17]. Finally, the fundamental frequency is scaled proportionally to the value of \( \alpha \) [17].

Once the MCEP and F0 features are converted (BAP features remain unchanged), speech resynthesis is performed to obtain waveforms with the converted speech. This is done using a vocoder and, depending on the goodness of the converted features and the vocoder, the resulting speech can be more or less natural and intelligible [18].

2.3. Search

The QbESDR system proposed in this work belongs to the family of pattern-matching techniques for search on speech. An overview is presented in Figure 4.

Previous work on QbESDR using gender conversion [16] relied on a large set of features plus feature selection for speech representation [24]. Nevertheless, as mentioned above, this paper aims to straightforwardly use the converted MCEP features for QbESDR and compare the performance with that achieved when extracting features from the converted waveforms. Therefore, there are two alternatives for speech representation: using...
MCEP features or obtaining a more complex representation that makes use of these features. Since raw acoustic features do not usually exhibit a good performance in QbESDR, Gaussian posteriorgrams were used for this purpose: a Gaussian posteriorgram represents each frame of a spoken utterance by means of a vector of dimension $G$: each element of this vector is the posterior probability of each of the $G$ Gaussians in a GMM given the frame. This representation was first proposed in [12] and used for QbESDR in [13, 14, 15], to cite some examples.

After feature extraction, given a query $Q = \{q_1, \ldots, q_n\}$ and a document $D = \{d_1, \ldots, d_m\}$ of $n$ and $m$ frames respectively, with vectors $q_i, d_j \in \mathbb{R}^G$ and $n \ll m$, DTW finds the best alignment path between these two sequences. Subsequence DTW [25] (S-DTW) was used in this system, since it allows the partial alignment of a short sequence (the query) with a longer sequence (the document). The first step consists in computing a cumulative cost matrix $M \in \mathbb{R}^{n \times m}$ for a given query and document as follows:

$$M_{i,j} = \begin{cases} c(q_i, d_j) & \text{if } i = 0 \\ c(q_i, d_j) + M_{i-1,0} & \text{if } i > 0 \\ c(q_i, d_j) + M^*(i,j) & \text{else} \end{cases}$$

(2)

where $c(q_i, d_j)$ is a function that defines the cost between query vector $q_i$ and document vector $d_j$, and

$$M^*(i,j) = \min \{M_{i-1,j}, M_{i-1,j-1}, M_{i,j-1}\}$$

(3)

In this paper, the log cosine similarity was used as the cost function as in [10] since it empirically showed a superior performance compared with other metrics:

$$\text{cost}(q_i, d_j) = -\log \frac{q_i \cdot d_j}{|q_i||d_j|}$$

(4)

This metric is normalized in order to turn it into a cost function defined in the interval $[0,1]$:

$$c(q_i, d_j) = \frac{\text{cost}(q_i, d_j) - \text{cost}_{\min}(i)}{\text{cost}_{\max}(i) - \text{cost}_{\min}(i)}$$

(5)

where $\text{cost}_{\min}(i) = \min_j \text{cost}(q_i, d_j)$ and $\text{cost}_{\max}(i) = \max_j \text{cost}(q_i, d_j)$.

After computing $M$, the S-DTW algorithm is used to find the best alignment path between $Q$ and $D$. According to this algorithm, the best alignment path ends at frame $b^*$:

$$b^* = \arg \min_{b \in 1,\ldots,m} M_{n,b}$$

(6)

Then, it is possible to backtrack the whole alignment path that starts at frame $a^*$.

A score must be assigned to each detection of a query $Q$ in a document $D$ in order to measure how likely the query is present in the document. In this system, the document is length-normalised by dividing the cumulative cost by the length of the warping path [26] and z-norm is applied afterwards [27].

3. Experimental framework

The evaluation framework used in this paper is that of the QbESDR task of Albayzin 2014 search on speech evaluation. It consists of a set of spoken documents extracted from TV broadcast news in Basque language under diverse background conditions [28]. The queries were recorded in an office environment, which serves to simulate a regular user querying a retrieval system via speech. Each query includes a basic and two additional examples from different speakers; in these experiments, only the basic example is used. Two different sets of queries are included in the dataset: development (dev) queries for parameter tuning and evaluation (eval) queries to assess system performance. Table 1 summarizes some statistics of the database.

<table>
<thead>
<tr>
<th>Data</th>
<th># recordings</th>
<th>Total</th>
<th>Min</th>
<th>Max</th>
<th># hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documents</td>
<td>1841</td>
<td>3 h 11 min</td>
<td>3.00 s</td>
<td>38.12 s</td>
<td>-</td>
</tr>
<tr>
<td>dev queries</td>
<td>100</td>
<td>2 min 51 s</td>
<td>1.35 s</td>
<td>2.29 s</td>
<td>772</td>
</tr>
<tr>
<td>eval queries</td>
<td>100</td>
<td>2 min 52 s</td>
<td>1.31 s</td>
<td>2.25 s</td>
<td>855</td>
</tr>
</tbody>
</table>

The evaluation metric used in this work to assess QbESDR performance is the maximum term weighted value [29], in accordance with the experimental protocol defined for Albayzin 2014 search on speech evaluation. This metric was adopted instead of actual TWV in order to ignore the performance loss caused by calibration issues.

4. Experiments and results

Before presenting the experimental results, some details of the different modules of the system described in Section 2 must be mentioned. The GMMs of the gender classification system were trained using the FA sub-corpus of Albayzin database [30], which includes around 4 hours of speech uttered by 200 different speakers (100 male, 100 female). The features used were 19 Mel-frequency cepstral coefficients (MFCCs) augmented with energy, delta and acceleration coefficients, and only voiced frames were considered. The number of mixtures of the GMMs was empirically set to 1024. The parameters $f_a, f_s$ and $k$ of the VC strategy were set to 700 Hz, 3000 Hz and 0.5, respectively, according to [17]. In the search stage, the silence intervals before and after the queries were automatically removed using the voice activity detection approach described in [31]. The number of Gaussians $G$ of the GMM used for Gaussian posteriorgram computation was empirically set to 128. It must be noted that the GMM of each experiment was trained with the features extracted from its corresponding documents, so they are different for each experiment.

The first experiment aimed at comparing system performance when extracting MCEP features from the converted waveforms (Synthesized) and when straightforwardly using the converted MCEP features (Converted). As shown in Figure 5, using the converted features leads to clearly better results for dev queries. In addition, experiments were run with different values of $|\alpha|$ in order to analyze the influence of this parameter in QbESDR results. The figure shows that the best results were obtained when converting male utterances to female voices with $|\alpha| = \pi/30$. The worst performance was achieved with $|\alpha| = \pi/12$ since such a coarse conversion leads to more distorted speech according to [17]. Results with $|\alpha| = \pi/36$ are worse than those obtained with $|\alpha| = \pi/30$ because the
conversion in this case is so subtle that it barely compensates the gender variability [17].

The experimental results on the dev queries were used to select the best target gender and $|\alpha|$ for the Synthesized and Converted systems: the best $|\alpha|$ was $\pi/30$ in both cases; the best target gender was male for the Synthesized experiment and female for the Converted experiment. Once parameter tuning was done, experiments were performed on the eval queries. Table 2 shows the results on the eval queries with the original speech (Original), with the Converted and Synthesized features, and when using an adapted version of the gender compensation strategy proposed in [16] (Queries only). The latter strategy consists in generating an opposite-gender version of each query and using the male (female) queries to search within the documents spoken by males (females). The results show that the Converted system achieves a relative improvement by 17% over the original speech. Performance was also evaluated for female queries with female documents (Fq-Fd), male queries with male documents (Mq-Md), female queries with male documents (Fq-Md) and male queries with female documents (Mq-Fd). The results show that the Converted system achieved the best performance for all the experiments. The improvement in Fq-Md and Mq-Fd experiments result from the gender variability reduction obtained with the proposed technique. Also, in the results, the gender distribution of the queries and documents for each gender was evaluated. In addition, the improvement of the results when using original speech (i.e. Fq-Fd) suggest that the GMM trained with both original and converted features leads to more robust Gaussian posteriorgrams. The table shows degraded performance of the Synthesized system in all the experimental conditions caused by the quality of the features extracted from the resynthesized recordings. The Queries only system exhibits almost the same performance as the Original one.

Further experiments were performed in order to find out whether the variability of the recordings is reduced using the proposed technique. For this purpose, an experiment was run inspired in a state-of-art speaker verification technique: i-vectors were extracted from the Original and Proposed spoken documents, and PLDA scoring [32] of all the pairs of documents was computed. This resulted in a mean score of -5.34 and standard deviation of 11.45 for Original documents, and a mean score of 0.84 and standard deviation of 9.71 for Proposed documents.

This suggests that the speakers of the documents are more similar to each other when applying the proposed gender variability compensation strategy.

5. Conclusions and future work

This paper analyzed the effect of reducing gender variability via voice conversion for QbESDR. Given a target gender, all the spoken documents and queries of the opposite gender are converted in order to make them sound as spoken by the target gender. In addition, in order to alleviate the negative effects of resynthesis, the converted MCEP features were straightforwardly used to obtain Gaussian posteriorgrams of the queries and documents to perform QbESDR using a DTW-based approach. The experimental framework of the QbESDR task in AlBayzin 2014 search on speech evaluation was used for validation, which consisted in a set of documents and queries in Basque language. The experiments showed a relative improvement by 17% with the proposed technique compared to the QbESDR results with the original documents and queries.

QbESDR results were analyzed according to the gender of the documents and queries, and the proposed method showed an improvement both in original and converted utterances. This suggests that a GMM trained with both original and converted features leads to more robust Gaussian posteriorgrams. In future work, the use of voice conversion for data augmentation in this scenario will be experimented.

The experimental validation showed a clear improvement when using converted features compared to extracting new features from the converted waveforms, which can be caused by the distortion introduced by the vocoder. Recent strategies for speech generation such as WaveNet have emerged in the voice conversion field, and it would be interesting to analyze the effect of different vocoders for QbESDR in the future. Also, the use of deep learning techniques for noise robust voice conversion will be assessed.

6. Acknowledgements

This work has received financial support from i) “Ministerio de Economía y Competitividad” of the Government of Spain and the European Regional Development Fund (ERDF) under the research projects TIN2015-64282-R and TEC2015-65345-P, ii) Xunta de Galicia (projects GPC ED431B 2016/035 and GRC 2014/024), and iii) Xunta de Galicia - “Consellería de Cultura, Educación e Ordenación Universitaria” and the ERDF through the 2016-2019 accreditations ED431G/01 (“Centro singular de investigación de Galicia”) and ED431G/04 (“Agrupación estratégica consolidada”).

Table 2: Eval results for all documents and queries and for different combinations of male (M) and female (F) documents (d) and queries (q) with different systems. Conversion parameters were tuned on dev queries.

<table>
<thead>
<tr>
<th></th>
<th>All</th>
<th>Fq-Fd</th>
<th>Mq-Md</th>
<th>Fq-Md</th>
<th>Mq-Fd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>0.1659</td>
<td>0.2753</td>
<td>0.1623</td>
<td>0.1334</td>
<td>0.1969</td>
</tr>
<tr>
<td>Converted</td>
<td>0.1937</td>
<td>0.2958</td>
<td>0.1958</td>
<td>0.2013</td>
<td>0.2249</td>
</tr>
<tr>
<td>Synthesized</td>
<td>0.0835</td>
<td>0.2042</td>
<td>0.0280</td>
<td>0.1672</td>
<td>0.0272</td>
</tr>
<tr>
<td>Queries only</td>
<td>0.1684</td>
<td>0.2753</td>
<td>0.1623</td>
<td>0.1490</td>
<td>0.1841</td>
</tr>
</tbody>
</table>

Figure 5: Results on the dev queries when parameters extracted from converted waveforms (Synthesized) and when using the converted parameters (Converted) for female (F) and male (M) target genders.
7. References


A Recurrent Neural Network Approach to Audio Segmentation for Broadcast Domain Data

Pablo Gimeno, Ignacio Viñals, Alfonso Ortega, Antonio Miguel, Eduardo Lleida
ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain

Abstract

This paper presents a new approach for automatic audio segmentation based on Recurrent Neural Networks. Our system takes advantage of the capability of Bidirectional Long Short Term Memory Networks (BLSTM) for modeling temporal dynamics of the input signals. The DNN is complemented by a resegmentation module, gaining long-term stability by means of the tied-state concept in Hidden Markov Models. Furthermore, feature exploration has been performed to best represent the information in the input data. The acoustic features that have been included are spectral log-filter-bank energies and musical features such as chroma. This new approach has been evaluated with the Albayzín 2010 audio segmentation evaluation dataset. The evaluation requires to differentiate five audio conditions: music, speech, speech with music, speech with noise and others. Competitive results were obtained, achieving a relative improvement of 15.75% compared to the best results found in the literature for this database.

Index Terms: audio segmentation, recurrent neural networks, LSTM, broadcast data

1. Introduction

Due to the increase of multimedia content and the generation of large audiovisual repositories, the need for automatic systems that can analyze, index and retrieve information in a fast and accurate way is becoming more and more important. Given an audio signal, the goal of audio segmentation is to obtain a set of labels in order to separate that signal into homogeneous regions and classify them into a predefined set of classes, e.g., speech, music or noise. This task is also important in other applications of speech technologies, such as automatic speech recognition (ASR) or speaker diarization, where an accurate labeling of audio signals can improve the performance of these systems in real-world environments.

Audio segmentation systems can be divided into two main groups depending on how the segmentation is performed: segmentation-and-classification systems and segmentation-by-classification systems.

• Segmentation-and-classification: these systems perform the segmentation task in two steps. First, boundaries that separate segments belonging to different classes are detected using a distance metric. Then the system classifies each delimited segment in a second step. Several distance metrics have been proposed in the literature: Bayesian Information Criterion (BIC) [1] [2], Generalized Likelihood Ratio (GLR) [3], or the Kullback Leibler (KL) distance [4] are some examples.

• Segmentation-by-classification: in this group of systems the segmentation is produced directly as a sequence of decisions over the input signal. A set of well-known machine learning classification techniques have been used in this task with good results, such as Gaussian Mixture Models (GMM) [5], Neural Networks (NN) [6] [7], Support Vector Machines (SVM) [8], or decision trees [9]. A factor analysis approach is proposed in [10], where a compensation matrix is computed for each class. In both approaches to audio segmentation it is usual that the original segmentation boundaries are refined by a resegmentation model. In this way, sudden changes in the labels can be prevented. Some resegmentation strategies rely on hidden Markov models [11] or smoothing filters [12].

The segmentation task is specially challenging when dealing with broadcast domain content because such documents contain different audio sequences with a very heterogeneous style. Different speech conditions and domains can be found, from telephonic quality to studio recordings to outdoors speech with different noises overlapped. Background music and a variety of acoustic noise effects are likely to appear as well. In this context, the technological evaluations Albayzín incorporated a segmentation task for broadcast news environments in 2010 [13]. This is the task we are focusing on in this paper.

The remainder of the paper is organized as follows: a theoretical background on LSTM networks and its applications is introduced in section 2. Section 3 describes our novel RNN-based segmentation system. Section 4 briefly introduces the experimental setup, describing the Albayzín 2010 evaluation dataset and presents all the experimental results obtained with our system. Finally, a summary and the conclusions are presented in section 5.

2. LSTM networks

Neural Networks are a powerful modeling tool for non linear dependencies that, since the early 2010s, have been increasingly applied to speech technologies [14] [15] [16]. One of the main disadvantages of traditional feed-forward neural networks when dealing with temporal series of information is that they process every example independently. However, Recurrent Neural Networks (RNNs) are able to capture temporal dependencies introducing feedback loops between the input and the output of the neural network. The long short term memory (LSTM) [17] networks are a special kind of RNN with the concept of memory cell. This cell is able to learn, retain and forget information [18] in long dependencies.
This capability becomes very useful to carry out long and short term analysis simultaneously. LSTM networks have been modified combining two of them in a Bidirectional LSTM (BLSTM) network. One processes the sequence in the forward direction while the other one processes the sequence backwards. This way the network is able to model causal and anticausal dependencies for the same sequence.

LSTM and BLSTM networks have been successfully applied to sequence modeling tasks in speech technologies such as ASR [19] [20], language modeling [21], speaker verification in text-dependent systems [22] or machine translation [23].

3. System description

Our proposed system is based on the use of RNNs to classify each frame in the audio signal. We have opted for a segmentation-by-classification solution, combining the RNN with a resegmentation module to get smoothed segmentation hypotheses. The three different blocks of our segmentation system (feature extraction, an RNN based classifier, and the final resegmentation module) are described below.

3.1. Feature extraction

The main features for the neural network consist of log Mel filter bank energies and the log-energy of each frame. Additionally, we combine Mel features with chroma features [24], due to its capability to capture melodic and harmonic information in music while being robust to changes in tone or instrumentation. All these features are computed every 10 ms using a 25 ms window. In order to take into account the dynamic information of the audio signal, first and second order derivatives of the features are computed. Finally, feature mean & variance normalization is applied.

3.2. Recurrent Neural Network

This is the core of our segmentation system. Its task is to classify each audio frame as belonging to one of the predefined acoustic classes. The neural architecture proposed can be observed in Fig. 1. As shown, it is composed by two stacked BLSTM layers with 256 neurons each. The outputs of the last BLSTM layer are then independently classified by a linear perceptron, which shares its values (weights and bias) for all time steps. Training and evaluation are performed with limited length sequences (3 seconds, 300 frames), limiting the delay of dependencies to take into account. However, the neural network emits a segmentation label per every frame processed at the input (every 10 ms, in our case).

The neural network has been trained using exclusively the train subset of the Albayzin 2010 database [13], which consists of 58 hours of audio sampled at 16 KHz. However, 15% of the train subset will be reserved for validation, which makes a total of 49 hours of audio for training and 9 hours for validation. Adaptive Moment Estimation (Adam) optimizer is chosen due to its fast convergence properties [25]. Data will be shuffled in each training iteration seeking to improve model generalization capabilities.

Despite our system consisting of an additional resegmentation module, the RNN itself is able to emit a segmentation hypothesis. This way, we can say that the first two blocks of our system can fully work as a segmentation system. All the neural architectures in this paper have been evaluated using the PyTorch toolkit [26].

3.3. Resegmentation module

RNNs have high capabilities to model temporal dependencies but its output may contain high frequency transitions which are unlikely to occur in signals which have a high temporal correlation such as human speech or music. With the objective of avoiding sudden changes in the segmentation process, we incorporate a resegmentation module to our system. In our case, it is implemented as a hidden Markov model where each acoustic class is modeled through a state in the Markov chain. Every state is represented by a multivariate Gaussian distribution with full covariance matrix. This block gets as input the pseudo log-likelihood for each class from the neural network and its corresponding segmentation hypotheses. No more a priori information is required because statistical distributions are estimated using the hypothesized labels for each file.

Input information is given every 10 ms, which can result in a noisy estimation of class boundaries. In order to reduce temporal resolution, scores are down-sampled by a factor $L$ using an $L$ order averaging zero-phase FIR filter to avoid distorting phase components [27]. First and second order derivatives of the scores are taken into account when computing the resegmentation. Additionally, each of the states in the Markov chain will consist of a left-to-right topology of a number $N_{ts}$ of tied states sharing the same statistical distribution. This way, by modifying $N_{ts}$ we can force the minimum length of a segment before a change in the acoustic class happens. Taking all this information into account, this minimum segment length forced by our resegmentation module can be computed as follows:

$$T_{min} = T_s LN_{ts}$$

where, $T_s$ is the sampling period of the neural network output (10 ms in our case), $L$ is the down-sampling factor and $N_{ts}$ is the number of tied states used in the Markov model.
4. Experimental setup and results

4.1. Database and metric description

The database consists of broadcast news audio in Catalan. The full database includes 87 hours of audio sampled at 16 KHz and divided in 24 files. The database was split into two parts: two thirds of the total amount of data are reserved for training, while the remaining third is used for testing. Five acoustic classes were defined for the evaluation. The classes are distributed as follows: 37% for clean speech (sp), 5% for music (mu), 15% for speech over music (sm), 40% for speech over noise (sn) and 3% for others (ot). The class “others” is not evaluated in the final test. A more detailed description of the Albayzín 2010 audio segmentation evaluation can be found in [13].

The main metric we will be using for evaluating our results is the Segmentation Error Rate (SER), inspired by the NIST metric for speaker diarization [28]. This metric can be interpreted as the ratio between the total length of the incorrectly labeled audio and the total length of the audio in the reference. Given the dataset to evaluate $\Omega$, each document is divided into continuous segments and the segmentation error time for each segment $n$ is defined as:

$$ \Xi(n) = T(n)[\max(N_{ref}(n), N_{sys}(n)) - N_{correct}(n)] $$  \hspace{1cm} (2)

where $T(n)$ is the duration of the segment $n$, $N_{ref}(n)$ is the number of reference classes that are present in segment $n$, $N_{sys}(n)$ is the number of system classes that are present in segment $n$ and $N_{correct}(n)$ is the number of reference classes that are present in segment $n$ and were correctly assigned by the segmentation system. This way, the SER is computed as follows:

$$ \text{SER} = \frac{\sum_{n \in \Omega} \Xi(n)}{\sum_{n \in \Omega} (T(n)N_{ref}(n))} $$  \hspace{1cm} (3)

Alternatively, the metric originally proposed for the Albayzín 2010 evaluation will also be taken into account. This metric represents the relative error averaged over all the acoustic classes:

$$ \text{Error} = \frac{\text{dur}({\text{miss}_i}) + \text{dur}({\text{fa}_i})}{\text{dur}({\text{ref}_i})} $$  \hspace{1cm} (4)

where $\text{dur}({\text{miss}_i})$ is the total duration of all miss errors for the $i$th acoustic class, $\text{dur}({\text{fa}_i})$ is the total duration of all false alarm errors for the $i$th acoustic class, and $\text{dur}({\text{ref}_i})$ is the total duration of the $i$th acoustic class according to the reference. A collar of ±1s around each reference boundary is not scored in both cases, SER and average class error, to avoid uncertainty about when an acoustic class begins or ends, and to take into account inconsistent human annotations.

4.2. Experimental results

For the experimental evaluation of our system, different front-end configurations were assessed. The starting point of our feature space exploration consists of a simple 32 log Mel filter bank. Our next step was increasing the frequency resolution by using a higher number of analysis bands in the filter bank, testing 64, 80 and 96 bands. Chroma features were incorporated in order to help our system to discriminate classes that contain music. Eventually, first and second order derivatives were computed to take into account dynamic information in the audio signal.

Results obtained on the test partition for the different front-end configurations using our BLSTM segmentation-by-classification system are presented in Table 1 in terms of SER and the Albayzín evaluation metric. When the number of analysis bands is increased, we can appreciate that the SER decreases, reaching its minimum using 80 bands. However, it can also be seen that using a higher number of bands can affect the system performance. This is the case of the 96 bands configuration, that increases its error compared to the 80 bands configuration. We can notice that, by incorporating chroma features, the error in the class “Speech over music” decreases significantly when compared to the 80 Mel coefficient configuration, with a relative improvement of 12.10%. This is due to the capabilities of chroma features to capture musical dependencies, which helps our system discriminate this class in a more accurate way. The best result for this set of experiments is obtained using the first and second order derivatives of the log Mel filter bank and the chroma features, achieving a SER of 15.91%, which is equivalent to an average class error of 25.84%. The performance of our segmentation system using the resegmentation module is evaluated in the following set of experiments.

Aiming to illustrate how the system performance is influenced by the inertia imposed by the resegmentation module, Fig. 2 shows the scatter plot of the relative improvement in performance versus the minimum segment length ($T_{\min}$) for different values of the down-sampling factor $L$. It can be seen that configurations that perform better have a minimum segment length between 0.5 and 1.5 seconds, which is in the order of magnitude of the 2 seconds collar applied in the evaluation.

The results on the test partition of the full segmentation system combining the BLTsMs and the resegmentation module for the best front-end configuration evaluated (80 Mel + chroma + derivatives) and for different values of the down-sampling factor $L$, and the minimum segment length $T_{\min}$, are shown in Table 2 in terms of SER and the Albayzín evaluation metric. If we compare the best result in this Table with the best result in Table 1, it can be seen that, by incorporating

<table>
<thead>
<tr>
<th>Feats</th>
<th>SER</th>
<th>Class Error(%)</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mu</td>
<td>sp</td>
<td>sm</td>
</tr>
<tr>
<td>32 Mel</td>
<td>16.09</td>
<td>30.89</td>
<td>33.07</td>
</tr>
<tr>
<td>64 Mel</td>
<td>17.98</td>
<td>31.45</td>
<td>31.38</td>
</tr>
<tr>
<td>80 Mel</td>
<td>18.07</td>
<td>30.81</td>
<td>31.48</td>
</tr>
<tr>
<td>96 Mel</td>
<td>17.33</td>
<td>30.81</td>
<td>31.48</td>
</tr>
<tr>
<td>80 Mel + Chr</td>
<td>16.14</td>
<td>30.13</td>
<td>31.66</td>
</tr>
<tr>
<td>80 Mel + Chr + Δ + ΔΔ</td>
<td>15.91</td>
<td>26.28</td>
<td>28.82</td>
</tr>
</tbody>
</table>

Table 1: SER, error per class and average error for BLSTM segmentation-by-classification system on the test partition for different feature configurations (Mel: log Mel filter bank, Chr: chroma, Δ + ΔΔ: 1st and 2nd order derivatives)
it can be seen that our BLSTM-HMM system performs better in all the acoustic classes. This error reduction is equivalent to a relative improvement of 15.24% in terms of SER and a 15.75% in terms of the average class error. The difference in the class “speech over music” is specially significant (23.60% vs 18.82%) with a relative improvement of 20.25%.

Table 3: Results obtained on the Albayzín 2010 test partition for different systems proposed in the literature compared to our BLSTM-HMM system

<table>
<thead>
<tr>
<th>System</th>
<th>SER</th>
<th>Class Error(%)</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>mu</td>
<td>sp</td>
</tr>
<tr>
<td>Eval winner [29]</td>
<td>19.30</td>
<td>19.20</td>
<td>39.50</td>
</tr>
<tr>
<td>FA HMM [10]</td>
<td>14.70</td>
<td>18.80</td>
<td>23.70</td>
</tr>
<tr>
<td>BLSTM-HMM</td>
<td>12.46</td>
<td>14.19</td>
<td>22.14</td>
</tr>
</tbody>
</table>

5. Conclusions

A new approach for audio segmentation based on RNNs is presented in this paper proving the capabilities of this kind of models in the audio segmentation task, achieving the best result so far in the Albayzin 2010 database. A segmentation-by-classification scheme has been followed, combining a classification system, which is mainly made of 2 BLSTM layers, with an smoothing back-end implemented through a Hidden Markov Model. Several front-end configurations were evaluated, proving the capabilities of chroma features for capturing musical structures when compared to a perceptual Mel filter bank. The combination of BLSTM and HMM has been proven to be appropriate, reducing significantly the system error by forcing a minimum segment length for the segmentation labels. Competitive results have been obtained with this new approach, resulting in a relative improvement of 15.75% when compared to the best result in the literature so far.

Regarding our contributions, front-end configuration seems to have a big impact in this task, specially when classifying classes that contain music. Just by modifying the input features we have achieved a significant improvement in the performance of our system. Furthermore, the introduction of RNNs in the audio segmentation task has been proven to be successful, improving the results obtained so far with traditional statistical models such as GMM/HMM or factor analysis. In future work we intend to improve even more these results by introducing more complex neural architectures after the BLSTM layers.

6. Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.
7. References


Improving Transcription of Manuscripts with Multimodality and Interaction

Emilio Granell, Carlos-D. Martínez-Hinarejos, Verónica Romero

Pattern Recognition and Human Language Technology Research Center, Universitat Politècnica de València, Camí de Vera s/n, 46022, València, Spain
{egranell, cmartine, vromero}@dsic.upv.es

Abstract

State-of-the-art Natural Language Recognition systems allow transcribers to speed-up the transcription of audio, video or image documents. These systems provide transcribers an initial draft transcription that can be corrected with less effort than transcribing the documents from scratch. However, even the drafts offered by the most advanced systems based on Deep Learning contain errors. Therefore, the supervision of those drafts by a human transcriber is still necessary to obtain the correct transcription. This supervision can be eased by using interactive and assistive transcription systems, where the transcriber and the automatic system cooperate in the amending process. Moreover, the interactive system can combine different sources of information in order to improve their performance, such as text line images and the dictation of their textual contents.

In this paper, the performance of a multimodal interactive and assistive transcription system is evaluated on one Spanish historical manuscript. Although the quality of the draft transcriptions provided by a Handwriting Text Recognition system based on Deep Learning is pretty good, the proposed interactive and assistive approach reveals an additional reduction of transcription effort. Besides, this effort reduction is increased when using speech dictations over an Automatic Speech Recognition system, allowing for a faster transcription process.

Index Terms: speech recognition, human-computer interaction, handwriting recognition, assistive transcription, deep learning

1. Introduction

Many documents used every day include handwritten text and, in many cases, such as for detecting fraudulent bank checks [1], it would be interesting to recognise these text images automatically. However, state-of-the-art handwritten text recognition (HTR) systems can not suppress the need of human supervision when high quality transcriptions are needed [2, 3, 4, 5].

A way of taking advantage of the HTR system is to combine it with the knowledge of a human transcriber, constituting the so-called Computer Assisted Transcription of Text Images (CATTI) scenario [6]. In this framework, the HTR system and the transcriber cooperate interactively to obtain the perfect transcription of the text line images. At each interaction step, the system uses the text line image and a previously validated part (prefix) of its transcription to propose an improved output. Then, the user finds and corrects the next system error, thereby providing a longer prefix which the system uses to suggest a new, hopefully better continuation. Moreover, the accuracy of the interactive system can be improved by providing it with additional sources of information, such as the speech dictation of the handwritten text over an automatic speech recognition (ASR) system.

In previous related works, multimodal combination was used to integrate the transcriber feedback into the main stream of information for word correction, by using on-line handwriting and speech [7, 8, 9]. In this work, we explore the idea of using speech dictation for feeding the interactive system with an additional source of information of the full text to transcribe.

The rest of the paper is organised as follows: Section 2 presents our multimodal proposal; Section 3 introduces the experimental framework; Section 4 explains the performed experiments and the obtained results; finally, Section 5 offers the conclusions and future work lines.

2. Multimodal Computer Assisted Transcription of Text Images

This section presents our proposal, which is composed of two parts, multimodal recognition and interaction.

2.1. Multimodal Recognition Framework

The natural language recognition problem aims to recover the text represented in an input signal. In the case of HTR, this input signal is usually a segmented line of a digitalised handwritten document [10]. Then, given a handwritten text line image or a speech signal represented by a feature vector sequence \( x = (x_1, x_2, \ldots, x_n) \), the problem for HTR and ASR is finding the most likely word sequence \( \hat{w} \) [2], that is:

\[
\hat{w} = \arg \max_{w \in W} P(w | x) = \arg \max_{w \in W} P(x | w) P(w)
\]

where \( W \) denotes the set of all permissible sentences, \( P(w) \) is the probability of \( w = (w_1, w_2, \ldots, w_n) \) approximated by the language model (usually \( n \)-gram) [11], and \( P(x | w) \) is the probability of observing \( x \) by assuming that \( w \) is the underlying word sequence for \( x \), evaluated by the optical or acoustical models for HTR and ASR, respectively. For both modalities, the state-of-the-art morphological models are based on deep neural networks [3, 12].

In this work, the search (or decoding) of \( \hat{w} \), for both modalities, was performed by using the EESEN decoding method [13], which is based on Weighted Finite State Transducers (WFST). The main reason for using this decoding method is that it allows obtaining not only a single best hypothesis, but also a huge set of best hypotheses compactly represented into a unimodal word lattice, in systems with morphological models based on neural networks.

To obtain the multimodal final output lattice, the lattices generated by the unimodal systems can be combined by removing the total cost of all paths from the unimodal lattices and by doing a union of the reweighted lattices [14]. The architecture of our multimodal recognition framework is presented in Figure 1, that shows that the output is a word lattice (more details in Section 3.3).
2.2. Multimodal and Interactive Framework

In the CATTI framework the transcriber is involved in the transcription process, since he/she is responsible for validating and/or correcting the system hypothesis. The system takes into account the handwritten text image and the transcriber feedback in order to improve the proposed hypotheses [6]. An example of a CATTI operation is shown in Figure 2. In this example, in the traditional post-edition approach, a transcriber should have to correct about two errors from the recognised hypotheses [6]. An example of a CATTI operation is shown in Figure 2. In this example, in the traditional post-edition approach, a transcriber should have to correct about two errors from the recognised hypotheses [6].

Formally, in the traditional CATTI framework [6], the system uses a given feature sequence, $x_{htr}$, representing a handwritten text line image and a user validated prefix $p$ of the transcription. In this work, in addition to $x_{htr}$, a sequence of feature vectors $x_{asr}$, which represents the speech dictation of the textual contents of the text line image, is used to improve the system performance. Therefore, the CATTI system should try to complete the validated prefix by searching for a most likely suffix $\hat{s}$ taking into account both sequences of feature vectors.

Following the assumptions presented in [15], the CATTI problem can be formulated as:

$$\hat{s} = \arg \max_s P(x_{htr} \mid p, s) \cdot P(x_{asr} \mid p, s) \cdot P(s \mid p)$$  \hspace{1cm} (2)

where the concatenation of $p$ and $s$ is $\omega$. As in conventional HTR and ASR, $P(x_{htr} \mid p, s)$ and $P(x_{asr} \mid p, s)$ can be approximated by morphological models and $P(s \mid p)$ by a language model conditioned by $p$. Therefore, the search must be performed over all possible suffixes of $p$ [6].

This suffix search can be efficiently carried out by using lattices [6] obtained from the combination of the HTR and ASR recognition outputs. In each interaction step, the decoder parses the validated prefix $p$ over the lattice and then continues searching for a suffix which maximises the posterior probability according to Equation (2). This process is repeated until a complete and correct transcription of the input text line image is obtained.

3. Experimental Framework

This section presents the datasets, the preprocess, the models, the system setup, and the evaluation metrics used in the experiments.

3.1. Datasets

The datasets used in this work correspond to a Spanish historical manuscript, a Spanish phonetic corpus, and a set of speech samples provided by five different native Spanish speakers.

3.1.1. Historical Manuscript: The Cristo Salvador Corpus

The Cristo Salvador corpus is a 19th century Spanish manuscript provided by Biblioteca Valenciana Digital (BiValDi), and it is publicly available for research purposes on the website of the Pattern Recognition and Human Language Technology (PRHLT) research center. It is a single writer book composed of 53 pages (the page 41 is presented in Figure 3)

\[\text{https://www.prhlt.upv.es}\]
that were manually divided into lines (such as the line shown at the top of Figure 4). This corpus presents some problematic image features, such as smear, background variations, differences in bright, and bleed-through (ink that trespasses to the other surface of the sheet).

We followed the directives of the hard partition defined in previous works [16, 17]. The first 30 pages (602 text lines) were used for training the optical and language models, while the following 3 pages (78 text lines) were used for validation purposes. The test set was composed of the lines of the page 41 (24 lines, 222 words); this page was selected for being, according to preliminary error recognition results, a representative page of the whole test set (the remaining 20 pages, 473 lines). This corpus contains 1213 lines, with a vocabulary of 3451 different words, and a set of 92 different characters, taking into account lowercase and uppercase letters, numbers, punctuation marks, special symbols, and blank spaces.

3.1.2. Speech Dataset: Albayzin and Cristo Salvador

The Spanish phonetic corpus Albayzin [18] was used for training the ASR acoustical models. This corpus consists of a set of three sub-corpus recorded by 304 speakers using a sampling rate of 16 KHz and a 16 bit quantisation. The training partition used in this work includes a set of 6800 phonetically balanced utterances, specifically, 200 utterances read by four speakers, 25 utterances read by 160 speakers, and 50 sentences read by 40 speakers with a total duration of about 6 hours. A set of 25 acoustical classes, 23 monophones, short silence, and long silence, was estimated from this corpus.

Test data for ASR was the product of the acquisition of the dictation of the contents of the lines of the page 41 by five different native Spanish speakers (i.e., a total set of 120 utterances, with a total duration of about 9 minutes) using a sample rate of 16 KHz and an encoding of 16 bits (to match the conditions of Albayzin data).

3.2. Preprocess and Feature Extraction

All the text line images were scaled to 64 pixels in height and a pre-processing was applied for correcting the slant and removing the background noise [19]. A text line image and the resulting image after the image preprocess are presented in Figure 4.

With respect to speech feature extraction, 39 Mel-Frequency Cepstral Coefficients composed of the first 12 cepstrals and log frame energy with first and second order derivatives were extracted from the audio files [20].

3.3. Models

Optical models are Convolutional Recurrent Neural Networks (CRNN), which consist of a convolutional and a recurrent blocks [21]. The convolutional blocks are composed of 3 convolutional layers of 16, 32, and 48 features maps. Each convolutional layer has kernel sizes of $3 \times 3$ pixels, horizontal and vertical strides of 1 pixel, LeakyReLU as activation function, and a maximum pooling layer with non-overlapping kernels of $2 \times 2$ pixels only at the output of the first two layers. Then, the recurrent blocks are composed of 3 recurrent layers. Each recurrent layer is composed of 256 Bidirectional Long-Short Term Memory (BLSTM) units. Finally, a linear fully-connected output layer is used after the recurrent block. Those models were trained using Laia [22].

Acoustical models were trained using EESEN [13]. This acoustical model is a Recurrent Neural Network (RNN) composed of 351 inputs for 9 neighbouring frames of cepstral features, 6 hidden layers with 250 BLSTM units, and an output layer with a softmax function [12].
The lexicon models for both modalities are in HTK lexicon format, where each word is modelled as a concatenation of characters for HTR or phonemes for ASR. The particularities of historical manuscripts, such as, writing style, epoch and subject, make it very difficult to find external resources that allow to improve the models. In general, a part of the book is used to train the models that are used to automatically transcribe the rest of the book. Therefore, the language model (LM) was estimated directly from the transcriptions of the pages included on the HTR training set using the SRILM ngram-count tool [23]. This language model is a 2-gram with Kneser-Ney back-off smoothing [24] interpolated with the whole lexicon in order to avoid out-of-vocabulary words, and it presents a perplexity of 742.8 for the test data.

3.4. System Setup

As previously stated, the decoding and lattice generation based on WFST for both modalities were implemented using the EESEN recogniser [13], however, the multimodal lattice combination was performed using lattice-combine from Kaldi [25]. In order to optimise the presented multimodal and interactive framework, the values of the main variables were set up on a validation set, as well as the limit of mouse actions for correcting each erroneous word on the interactive transcription experiments, that was set to 3 [6].

3.5. Evaluation Metrics

The quality of the transcriptions is assessed using the Word Error Rate (WER), which allows us to obtain a good estimation for the transcriber post-edition effort. The WER is based on the Levenshtein edit distance [26] and it can be defined as the minimum number of words that have to be substituted, deleted and inserted to transform the transcription into the reference text, divided by the number of words in the reference text.

The quality of the lattices can be defined as the quality of the best hypotheses contained in them, and it is known as oracle error rates. Then, the quality of the word lattices is estimated by the oracle WER, which represents the smaller WER that can be obtained from the word sequences contained in them.

The overall interactive performance is given by Word Stroke Ratio (WSR), which can also be computed by using the reference text. After each hypothesis proposed by the system, the longest common prefix between the hypothesis and the reference text is obtained and the first error from the hypothesis is corrected. This process is iterated until a full match is achieved. Therefore, the WSR can be defined as the number of user corrections that are necessary to produce correct transcriptions using the interactive system, divided by the total number of words in the reference text. This definition makes the WER comparable to the WSR. The relative difference between them gives us the effort reduction (EFR), which is an estimation of the reduction of the transcription effort that can be achieved by using the interactive system.

The statistical significance of the experimental results is estimated by means of p-values with a threshold of significance of $\alpha = 0.05$ that were calculated through the Welch t-test [27] using the statistical computing tool R [28].

4. Experimental Results

Table 1 presents the obtained experimental results. As it can be observed, in the post-edition results the quality of the lattices offered by the handwritten text recognition system is pretty good, concretely it presents a WER equal to 8.9% and an oracle WER equal to 1.8%. In this case, speech recognition does not seem to be a good substitute for handwriting recognition. The quality of the lattices obtained by the speech recognition system present a WER equal to 31.4% and an oracle WER equal to 8.5%.

Regarding multimodality, the quality of the lattices obtained from the lattice combination of both modalities presents a WER equal to 10.6%. However, these multimodal lattices presents an oracle WER equal to 0.8%. Even though the combination technique does not improve the unimodal HTR WER, it allows to reduce the oracle WER substantially. Therefore, an outstanding effect on interactive transcription can be expected, since the oracle WER is related to the quality of the alternatives offered by the interactive and assistive system (the lower the oracle WER, the better the alternatives).

Concerning the CATTI results, 4.1% of estimated interactive human effort (WSR) was required for obtaining the perfect transcription from the HTR lattices, which represents 53.9% of relative effort reduction (EFR) over the HTR baseline (WER equal to 8.9%, $p = .051$). On the other side, no effort reduction can be considered when only ASR is used at the input of the interactive system. However as expected, the multimodal combination not only represents 56.1% of relative improvement on the estimate interactive human effort (1.8% over 4.1%, $p = .091$), but these improvements are statistically significant when compared with the HTR baseline (EFR equal to 79.8%, $p < .001$).

5. Conclusions

In this paper, the use of multimodal combination for improving the CATTI system presented in previous works has been studied. Multimodal combination allows us to provide additional sources of information to the assistive transcription system, such as speech dictation of the textual contents of the document to transcribe.

The obtained results show that the combination technique used, even though it does not improve the best hypothesis offered by the unimodal HTR system, it may produce new bigrams that increase the search alternatives. Moreover, the adjustment of the word posterior probabilities can increase the probabilities of the correct words, reaching better hypotheses that allows the assistive transcription system to provide an additional and significant reduction of the human effort.

In future work, we will study the use of other combination techniques, the use of sentences in the handwritten text corpus instead of lines (in order to make multimodality more natural), and the use of the information of context given by the previous lines.

6. Acknowledgments

This work was partially supported by the following research projects: READ - 674943 (European Union’s H2020) and CoMUN-HaT - TIN2015-70924-C2-1-R (MINECO / FEDER).
7. References


Improving Pronunciation of Spanish as a Foreign Language for L1 Japanese Speakers with Japañol CAPT Tool

Cristian Tejedor-García¹, Valentín Cardeñoso-Payo¹, María J. Machuca², David Escudero-Mancebo¹, Antonio Ríos², Takuya Kimura³

¹Department of Computer Science, University of Valladolid, Spain
²Department of Spanish Philology, Autonomous University of Barcelona, Spain
³Department of Spanish Language and Literature, Seisen University, Japan

cristian@infor.uva.es

Abstract

Availability and usability of mobile smart devices and speech technologies ease the development of language learning applications, although many of them do not include pronunciation practice and improvement. A key to success is to choose the correct methodology and provide a sound experimental validation assessment of their pedagogical effectiveness. In this work we present an empirical evaluation of Japañol, an application designed to improve pronunciation of Spanish as a foreign language targeted to Japanese people. A structured sequence of lessons and a quality assessment of pronunciations before and after completion of the activities provide experimental data about learning dynamics and level of improvement. Explanations have been included as corrective feedback, comprising textual and audiovisual material to explain and illustrate the correct articulation of the sounds. Pre-test and post-test utterances were evaluated and scored by native experts and automatic speech recognition, showing a correlation over 0.86 between both predictions. Sounds [s], [ʃ], [ɾ] and [s], [ɾ], [θ] explain the most frequent perception and production failures, respectively, which can be exploited to plan future versions of the tool, including gamified ones. Final automatic scores provided by the application highly correlate (r>0.91) to expert evaluation and a significant pronunciation improvement can be measured.

Index Terms: Computer-Assisted Pronunciation Training, corrective feedback, pronunciation training, second language learning, speech recognition, speech synthesis, minimal pairs

1. Introduction

Steady advances in automatic speech recognition (ASR) and speech synthesis (TTS) lead to an increase of their use in Computer-Assisted Pronunciation Training (CAPT) tools [1]. Availability and usability of mobile smart devices motivates the development of foreign language learning applications [2, 3], some of them include pronunciation practice and improvement as the main focus. Although existing literature on CAPT indicates that these systems tell us how good learners perform and how to improve their pronunciation [4, 5], choosing the correct methodology and elaborating a sound experimental validation of the assessment of pedagogical effectiveness of these tools are essential to further develop the field [6, 7].

Japañol¹ is a CAPT tool we have specifically designed for native Japanese learners of Spanish as a foreign language. Recent studies suggest that these learners mispronounce vowels and consonants [8, 9], one of the most frequent errors being the substitution of Spanish round back high vowel [u] by Japanese unrounded central-back vowel [ʉ]. Consonant mispronunciations usually stem from different phonological processes in both languages: phoneme substitutions between allomorphic variants or changes in the place of articulation are the most common errors. For instance, [l] and [ɾ] are allophones of the same rhotic phoneme /tɾ/ in Japanese. By contrast, both sounds are different phonemes in Spanish. Therefore, the trill Spanish phoneme /tɾ/ is often pronounced by Japanese speakers as a flap ([ɾ]) or even as a lateral ([l]) [10].

Most frequent mistakes are found in fricatives: [f] is often confused with [x], especially when these sounds are followed by [u]. In this way, two different Spanish words “juego” (fire) and “juego” (play) can be pronounced in the same way by Japanese speakers ([ɾ] is realised as labiodental and /x/ as velar) and none of these sounds exist in Japanese. The only similar fricative phoneme is /h/, and its realization depends on the vocalic context, for example, [ç] appears before palatal vowel or semivowel (/i/ or /y/) and [θ], before velar one ([ɾ] or [θ]) is another fricative sound which causes pronunciation problems. The substitution of this phoneme by [s] is a very common mistake, a phenomenon known as Asian “seseo” (lispe) [11], although it should not be considered a characteristic of non-nativeness, since this pronunciation is accepted in Spanish American variants [12]. Finally, according to phonotactic rules, Spanish allows consonant followed by consonant in onset clusters (i.e., ‘prado’) and coda ones (i.e., ‘inscripción’). The combination of fricative /θ/ and /ɾ/ or /ɾ/ in onset also triggers mispronunciation. This problem is not only due to the fact that these consonants exhibit errors in simple onset position, but also because most of the Japanese syllables have a consonant followed by vowel structure.

In previous works, we presented the development of serious games for training foreign pronunciation based on minimal pairs tasks [13, 14]. We were able to assess user’s pronunciation level in a L2, that is, those with a certified higher level consistently reached better scores in the game [15]. Besides, we also have discussed that the introduction of corrective feedback [16] allowed us to confirm that there was pronunciation improvement among users after the first stages of use. However, continued use of the tool seemed to invariably lead to stagnation. Finally, while the freedom of movement on game-oriented tools lead users to boost their score by repeating those tasks they found easy, learning-oriented tools insist on users’ difficulties offering guided and corrective feedback and achieving better effectiveness and efficiency pedagogical results [17].
In this paper we present an empirical evaluation of effectiveness in pronunciation training in Japanese students of Spanish as a foreign language. Corrective feedback in pronunciation training of second language acquisition could be implicit or explicit, in terms of whether or not the learner is informed of the corrected form of the error [18]. Since it is recognized as an essential component of these kinds of systems, an explanation mode has been included, comprising textual and audiovisual material which provides explanations and illustrations on the correct articulation of the sounds included in each lesson. We also provide a guided path to users to follow in order to complete all activities, based on their results. The preferable characteristics of corrective feedback are: unambiguous, understandable, detectable, short and should preferably take account of learner characteristics, both proficiency and literacy level [19]. We provide different types of feedback depending on the tasks and results (see subsection 2.2).

In section 2, the experimental procedure is described, which includes the participants and protocol stages, the speech material processing and Japañol description. Results section shows users’ interaction degree with the CAPT system, the most difficult Spanish sounds found in perception and production tasks and user’s human rater and ASR scores consistency, correlation and improvement. We end the paper with a discussion about the relevance of the results and the conclusions and future work.

2. Experimental procedure

2.1. Protocol description

A total of eight native Japanese students between 20 and 22 years old were selected as participants for our experiment. All participants qualified for the same Spanish as foreign language for beginners (A2-B1) course at the Language Center of the University of Valladolid. In this way we guaranteed (1) that all students had the same initial level of Spanish and (2) that our experiment realistically reproduced the variety of persons that attend to these type of courses.

Systematization of Spanish mispronunciations produced by Japanese speakers was the first step to choose minimal pairs in Spanish [20]. A set of 56 words, chosen according to the pairs to be worked out along the game and selected according to their phonetic difficulty, where spoken by each of the 8 participants, both before (pre-test) and after (post-test) training sessions. Participants exclusively used the CAPT system in the three 45-minutes-maximum training sessions; with a delay of at least 48 hours. A total of 84 minimal pairs corresponding to mispronunciations were presented to participants, 12 in each of the seven lessons. Students were asked not to complete more than three lessons per session. The software application was installed under an Android emulator (NOX App player) in the computers of the Language Center multimedia laboratory. At the beginning of the first session, students were instructed on how to use the software. Then, they had to work individually and did not have any communication with either instructors or classmates. Each participant worked inside an individual cubicle equipped with a headset with microphone.

Once finished the post-test, a manual revision and isolation of the recorded words of pre and post-test was carried out in order to elaborate a perceptual listening test. Five expert phoneticians and native speakers assigned a correct/incorrect value to each word, plus some extra annotations about the pronuncia-

![Figure 1: Standard flow to complete a lesson in Japañol.](image-url)
words have their orthographic and phonetic transcription repre-
sentations. Users can also request listen to the target word again 
with a replay button. Speed varies alternately between slow and
normal speed rates. Finally, the system changes word color to
green (success) or red (failure) with a chime sound. Pronunci-
ation is the fourth mode (step 7) which aim is to produce as 
well as possible, both words, separately, of the five minimal
pairs presented with their phonetic transcription. Google’s ASR
determines automatically and in real time acceptable or non-
acceptable inputs. In each production attempt the tool displays 
a text message with the recognized speech, plays a right/wrong
sound and changes word’s color to green or red. The maximum
number of attempts per word is five in order not to discourage 
users. However, after three consecutive failures, the system of-
fers to the user the possibility of request a word synthesis as an 
explicit feedback as many times as they want with a replay but-
ton. Mixed mode is the last mode of each lesson (step 8). Nine
production and perception tasks alternate at random in order to
further consolidate obtained skills and knowledge.

3. Results

Table 1 shows user’s interaction degree with Japañol. After all
training sessions, Japanese learners spent an average of 100.12
minutes performing the proposed tasks. Users consumed a
83.06% of the mean effective time for carry out interactive exer-
cises in EXP, DIS, PRO and MIX modes. As a mean term, users 
listened to the TTS system 628.51 times and used the ASR sys-
tem 247.13 times, giving a rate of 8.74 uses of the TTS/ASR per
minute. Table 1 also shows important differences in the use of 
the tool depending on the user. For instance, the user who per-
formed tasks of the PRO mode in the fastest way spent 22.43
minutes performing the proposed tasks. Users consumed a
training sessions, Japanese learners spent an average of 100.12

seconds performing the proposed tasks. Users consumed a
83.06% of the mean effective time for carry out interactive exer-
cises in EXP, DIS, PRO and MIX modes. As a mean term, users 
listened to the TTS system 628.51 times and used the ASR sys-
tem 247.13 times, giving a rate of 8.74 uses of the TTS/ASR per
minute. Table 1 also shows important differences in the use of 
the tool depending on the user. For instance, the user who per-
formed tasks of the PRO mode in the fastest way spent 22.43
minutes performing the proposed tasks. Users consumed a
training sessions, Japanese learners spent an average of 100.12

Table 1: Events by user with the CAPT tool in the whole experiment. THE, EXP, DIS, PRO and MIX correspond to Theory, Exposure, 
Discrimination, Pronunciation and Mixed modes, respectively. π, m and M are the mean, minimum and maximum values. Time (min)
row represents the time spent in minutes per person in each mode in the whole experiment. #Tries is the number times a mode is 
practiced by each user. Mand. and Req. mean mandatory and requested TTS listenings. Productions use ASR.

<table>
<thead>
<tr>
<th></th>
<th>THE</th>
<th></th>
<th>EXP</th>
<th></th>
<th>DIS</th>
<th></th>
<th>PRO</th>
<th></th>
<th>MIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>π</td>
<td>m</td>
<td>M</td>
<td>π</td>
<td>m</td>
<td>M</td>
<td>π</td>
<td>m</td>
<td>M</td>
<td>π</td>
</tr>
<tr>
<td>Time (min)</td>
<td>16.96</td>
<td>11.13</td>
<td>20.75</td>
<td>18.73</td>
<td>12.77</td>
<td>21.90</td>
<td>7.26</td>
<td>6.03</td>
<td>8.93</td>
</tr>
<tr>
<td>#Tries</td>
<td>8.25</td>
<td>7</td>
<td>10</td>
<td>11.38</td>
<td>13.50</td>
<td>10</td>
<td>14</td>
<td>10.66</td>
<td>8</td>
</tr>
<tr>
<td>#Mand.listenings</td>
<td>-</td>
<td>-</td>
<td>308.75</td>
<td>220</td>
<td>390</td>
<td>94.75</td>
<td>80</td>
<td>134</td>
<td>-</td>
</tr>
<tr>
<td>#Req.listenings</td>
<td>-</td>
<td>-</td>
<td>97.5</td>
<td>70</td>
<td>126</td>
<td>35.0</td>
<td>0</td>
<td>82</td>
<td>43.13</td>
</tr>
<tr>
<td>#Recordings</td>
<td>-</td>
<td>-</td>
<td>64.6</td>
<td>47</td>
<td>81</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>#Discriminations</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>94.75</td>
<td>80</td>
</tr>
<tr>
<td>#Productions</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2: Confusion matrix of discrimination task-tokens (diagonal: right discrimination task-tokens). The rows are the sounds 
expected by the tool and the columns are the sounds selected by the user. Success refers to the success rate of the corresponding
sound row. # Lis is the number of requested listenings to the sound row.

<table>
<thead>
<tr>
<th>Discrimination task-tokens</th>
<th># Lis</th>
<th>[fl]</th>
<th>[f]</th>
<th>[l]</th>
<th>[rr]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[fl]</td>
<td>85</td>
<td>23</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[f]</td>
<td>20</td>
<td>57</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[l]</td>
<td>113</td>
<td>9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[rr]</td>
<td>29</td>
<td>6</td>
<td>14</td>
<td>92</td>
<td>-</td>
</tr>
<tr>
<td>[l]</td>
<td>6</td>
<td>18</td>
<td>92</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[rr]</td>
<td>27</td>
<td>0</td>
<td>19</td>
<td>45</td>
<td>19</td>
</tr>
<tr>
<td>[l]</td>
<td>20</td>
<td>0</td>
<td>109</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>[rr]</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>49</td>
<td>-</td>
</tr>
<tr>
<td>[l]</td>
<td>40</td>
<td>0</td>
<td>50</td>
<td>18</td>
<td>75.5</td>
</tr>
<tr>
<td>[rr]</td>
<td>30</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

results provide in first place the desired word. Users can try a
maximum of five wrong productions per word. We can observe 
a success improvement from first to last attempt (last column),
being the highest ones [fl] (48.4%) and [f] (27.7%) sounds. At
first attempt, the most confused pair in production tasks is [fl]-
[f] (88 times) and the least confused one is [l]-[rr] (20 times).
The sounds with the lowest discrimination success rate are [fl] 
and [s] (both < 35%), and those with the highest discrimination 
success rates are [l] and [f] (both > 75%). At last attempt, the 
most confused pair in production tasks is [s]-[f] (102 times) and 
the least confused one is [l]-[f] (3 times). The sounds with the
lowest production success rate are [s] and [f] (both < 65%). A 
higher number of requested listenings (first column) appears for
last attempt pronunciation at lower success rates. The highest
production rate success sounds are [l] and [f] (both > 96%).

Table 4 presents the scores for each user at any of the given
stages of the experiment (pre-test, CAPT tool, post-test, and a 
delta score of pre and post-test). EXP and ASR scores refer to 
both tests learners’ qualifications by human raters and Google
ASR, respectively. These scores are computed by summing up 
the number of correct words per speaker and normalizing the result
to the range [0,10]. JAP score is computed by the number of 
correct and incorrect task-tokens while doing the required task-
types of the training modes (Discrimination, Pronunciation 
and Mixed modes) as a qualification to rank the participants.

Concerning to EXP values, a consistency test among hu-
man raters based on the Fleiss’ Kappa statistical indicator was 
accomplished both for pre-test and post-test evaluations. For lo-
Table 3: Confusion matrix of pronunciation task-tokens at first and last attempt per word sequence (diagonal: right pronunciation task-tokens at first and last attempt per word sequence). The rows are the sounds expected by the tool and the columns are the sounds produced by the user. Success refers to the success rate of the corresponding sound row. #Ris is the number of requested listerings to the sound row.

<table>
<thead>
<tr>
<th>#Ris</th>
<th>(l)</th>
<th>(fr)</th>
<th>(r)</th>
<th>(s)</th>
<th>(0)</th>
<th>(f)</th>
<th>(u)</th>
<th>(xu)</th>
<th>Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>70</td>
<td>17</td>
<td>85</td>
<td>48</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td>26.2%</td>
</tr>
<tr>
<td>1</td>
<td>59</td>
<td>40</td>
<td>38</td>
<td>23</td>
<td>68</td>
<td></td>
<td></td>
<td></td>
<td>36.5%</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>11</td>
<td>2</td>
<td>12</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>60.3%</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>10</td>
<td>1</td>
<td>89</td>
<td>126</td>
<td>19</td>
<td>4</td>
<td></td>
<td>75.4%</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>4</td>
<td>8</td>
<td>84</td>
<td>124</td>
<td></td>
<td></td>
<td></td>
<td>71.8%</td>
</tr>
<tr>
<td>6</td>
<td>92</td>
<td>20</td>
<td>64</td>
<td>38</td>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td>34.5%</td>
</tr>
<tr>
<td>2</td>
<td>120</td>
<td>41</td>
<td>54</td>
<td>63</td>
<td>114</td>
<td>24</td>
<td>7</td>
<td></td>
<td>49.2%</td>
</tr>
<tr>
<td>0</td>
<td>9</td>
<td>24</td>
<td>8</td>
<td>40</td>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td>62.5%</td>
</tr>
<tr>
<td>3</td>
<td>62</td>
<td>35</td>
<td>58</td>
<td>22</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td>61.4%</td>
</tr>
<tr>
<td>2</td>
<td>29</td>
<td>19</td>
<td>14</td>
<td>34</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td>64.2%</td>
</tr>
</tbody>
</table>

Table 4: Different user scores in pre-test, post-test and game by experts, ASR and game in a scale of [0,10]. ID, EXP, ASR and JAP refer to user identifier, experts, Google ASR and Japañol, respectively.

<table>
<thead>
<tr>
<th>Pre-test</th>
<th>Game</th>
<th>Post-test</th>
<th>Δ (Pre/Post)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP</td>
<td>ASR</td>
<td>JAP</td>
<td>EXP</td>
</tr>
<tr>
<td>07</td>
<td>9.8</td>
<td>5.1</td>
<td>9.4</td>
</tr>
<tr>
<td>05</td>
<td>8.6</td>
<td>4.5</td>
<td>9.1</td>
</tr>
<tr>
<td>01</td>
<td>8.1</td>
<td>2.9</td>
<td>8.0</td>
</tr>
<tr>
<td>02</td>
<td>7.3</td>
<td>3.7</td>
<td>8.4</td>
</tr>
<tr>
<td>03</td>
<td>6.9</td>
<td>1.9</td>
<td>6.3</td>
</tr>
<tr>
<td>08</td>
<td>6.8</td>
<td>2.8</td>
<td>6.5</td>
</tr>
<tr>
<td>06</td>
<td>5.8</td>
<td>0.8</td>
<td>5.9</td>
</tr>
<tr>
<td>04</td>
<td>5.4</td>
<td>2.1</td>
<td>6.9</td>
</tr>
</tbody>
</table>

4. Discussion and Conclusion

We have described the experiment results with a CAPT system of native Japanese learners of Spanish as a foreign language, that include a software tool, Japañol, that we believe is worth taking into account when thinking about possible teaching complement. The results have shown that Japanese learners of Spanish have difficult with [l] in onset clusters position in both perception (Table 2) and production (Table 3). [s]-[f] present similar results, they tend to substitute [f] by [s], but this pronunciation is accepted in Latin American Spanish. About liquid consonants, Japanese speakers are more successful at phonetically producing [l] and [r] than discriminating these phonemes. Japanese speakers have already acquired these sounds since they are allophones of a same liquid phoneme in Japanese. For this reason, it does not seem to be necessary to distinguish them in Japanese, whereas it is in Spanish.

Applied methodology has proved coherent and efficient because easier tasks (exposure, discrimination) were presented before tougher ones (pronunciation) in time and repetition terms (Table 1). It has also led to better final scores (Table 4). Participants consistently resorted to TTS models when faced with difficulties both in perception and production modes (#Ris column of Tables 2 and 3 and #Ris column of Table 1). The fact that our CAPT tool makes use of ASR and TTS systems is the principal pedagogical and operational concern. The quality of synthetic voice in the rendering of minimal pairs appears to have been adequate. Judging from gathered data, the TTS system employed seems to be beneficial for students [22, 23]; the rate of success significantly increases after undertaking the exposure activities imposed by feedback. The role of ASR is even more crucial as it offers diagnosis to users as real-time automatic feedback. Nowadays, ASR systems have some limitations such as isolated or infrequent words, adaptation to technology and L1 transferred pronunciation utterances as correct L2 words (false positives). However, [24] demonstrated the effectiveness of ASR-based CAPT tool for training users in the production of decontextualized isolated words and [25] reported L2 French vowel/s/ production improvement after training with a mobile ASR system.

Human raters’ post-test scores fairly correlate with ASR and game ones, being useful in the future to be able to evaluate a large amount of users reducing human costs (Table 4). The lowest pre-test scores’ users improved more than the best ones. However, they did not reach better results than the top ones. This is due to the fact that the tool does not give extra activities when the limit is reached. As a future work, we are collaborating with Seisen University with a bigger group of students divided into experimental, control and in-classroom groups. We are also considering the possibility of applying the methodology to other more exotic and minority languages. Finally, a comparison between a game-oriented version versus a learning-oriented one could be our next step.
5. References


Exploring E2E speech recognition systems for new languages

Conrad Bernath1, Aitor Álvarez1, Haritz Arzelus1, Carlos-D. Martínez-Hinarejos2

1Human Speech and Language Technology Group, Vicomtech, Spain
2Pattern Recognition and Human Language Technologies Research Center, Universitat Politècnica de València, Spain

[cbernath,aalvarez,harzelus]@vicomtech.org, cmartine@dsic.upv.es

Abstract

Over the last few years, advances in both machine learning algorithms and computer hardware have led to significant improvements in speech recognition technology, mainly through the use of Deep Learning paradigms. As it was amply demonstrated in different studies, Deep Neural Networks (DNNs) have already outperformed traditional Gaussian Mixture Models (GMMs) at acoustic modeling in combination with Hidden Markov Models (HMMs). More recently, new attempts have focused on building end-to-end (E2E) speech recognition architectures, especially in languages with many resources like English and Chinese, with the aim of overcoming the performance of LSTM-HMM and more conventional systems.

The aim of this work is first to present the different techniques that have been applied to enhance state-of-the-art E2E systems for American English using publicly available datasets. Secondly, we describe the construction of E2E systems for Spanish and Basque, and explain the strategies applied to overcome the problem of the limited availability of training data, especially for Basque as a low-resource language. At the evaluation phase, the three E2E systems are also compared with LSTM-HMM based recognition engines built and tested with the same datasets.

Index Terms: speech recognition, deep learning, end-to-end speech recognition, recurrent neural networks

1. Introduction

Automatic Speech Recognition (ASR) systems have historically employed Hidden Markov Models (HMMs) to capture the time variability of the speech signal and Gaussian Mixture Models (GMMs) to model the HMM state probability distributions. Relevant advances in both machine learning and computational capacity have led to significant improvements in the field, mainly by means of Deep Learning algorithms. Thereby, numerous works have shown that Deep Neural Networks (DNNs) in combination with HMMs can outperform GMM-HMM systems at acoustic modeling on a variety of datasets [1].

Recently, new attempts have focused on building End-to-End (E2E) ASR architectures, especially in languages with many resources like English and Chinese [2]. These new architectures intend to benefit from the vast amount of new data available, and to make training a much more efficient process by allowing a better optimization of the system as a whole unit [3]. Besides, one of their most remarkable advantages is to build ASR engines without the need of a phoneme set or a pronunciation lexicon of the language. This definitely reduces the human effort to manually construct resources to help the system to estimate their components. This novel approach allows the system to extract the best features and to employ a single optimization criterion that leads to a better general performance.

The interest in building E2E ASR systems that directly map the input speech signal to grapheme/character sequences and that jointly train the acoustic, pronunciation and language models has exponentially grown in the last years [3, 4, 5, 6].

Two main approaches have been mainly employed to build E2E ASR models. The Connectionist Temporal Classification (CTC) is probably the most widely used criterion for systems based on characters [2, 7, 8], sub-words [9] or words [10]. It employs Markov assumptions and dynamic programming to efficiently solve sequential problems [2, 3, 7]. On the other hand, attention-based methods use an attention mechanism to perform alignment between acoustic frames and characters [4, 5, 6]. Unlike CTC, it does not require several conditional independence assumptions to obtain the label sequence probabilities, allowing extremely non-sequential alignments like in the case of machine translation.

Several techniques have been applied in the literature to enhance the performance of E2E ASR engines. These techniques are mainly focused on compensating for the lack of training data, adapting the characteristics of the system to some specific domains or on gaining robustness in different acoustic environments. Data augmentation aims to extend the training material by generating new synthetic data through noise injection or by augmenting the audios using Vocal Tract Length Perturbation (VTLP), tempo modification or speed alteration [11]. The Transfer Learning technique is the improvement of a new learning by taking advantage of the knowledge obtained from a previously learned related task [12]. For E2E ASR models, this technique consists of using the weights of a previously trained model in a particular language as initial weights for the training of a new target language. Fine Tuning is commonly applied when existing E2E models have to be adapted to some particular conditions. It can be seen as a subtype of Transfer Learning but without freezing layers or making changes over them.

A number of enhancement techniques have also been employed to improve the performance of these systems under acoustically noisy conditions. In addition to the front-end methods applied at feature level [13, 14] or training on noisy data with a range of SNR values [15], other methods such as Dropout [16] or Curriculum Learning [17] have also been proven to improve robustness against noise.

In this work, the construction and evaluation of several E2E systems are presented for English, Spanish and Basque, considering the latter a low resource language. In addition, several new modeling techniques explained above were applied with the aim of outperforming baseline E2E engines and exploring further possibilities within more challenging acoustic domains.
and languages. Finally, the results achieved were compared with those obtained by LSTM-HMM based systems through the use of the same training and evaluation datasets.

This paper is organized as follows. Section 2 describes the main architecture of the E2E systems developed within this work. Section 3 describes the training and evaluation data employed, whilst Section 4 presents the baseline ASR systems built for each language including both E2E and LSTM-HMM architectures. The experiments performed are described in Section 5 and finally, Section 6 draws conclusions and looks at the future work.

2. System overview

All the E2E systems presented in this work were developed following the Deep Speech 2 architecture [2]. The core of the system is basically an RNN model, in which speech spectrograms are ingested and text transcriptions are provided as output. Although the Long-short-term Memory (LSTM) is widely used as RNN model, in this architecture Gated Recurrent Units (GRU) [18] are employed, since they have been proven to be trained more rapidly and to be less likely to diverge [19].

A sequence of 2 layers of 2D convolutional neural networks (CNN) are employed as spectral feature extractor from spectrograms. The first layer was composed of 1 input and 32 output channels and it uses filters of size 41 \times 11 and stride of size 2 \times 2. The second layer takes as input the output of the first layer, composed of 32 channels. The output of the second layer incorporates 32 channels as well. This second layer employs a filter dimension of 21 \times 11 and stride of size 2 \times 1. A 2D batch normalization function is applied to the output of both layers, in addition to a hard tanh function as an activation function.

The E2E systems are set up using 5 layers of bidirectional GRU layers. Each hidden layer is composed of 800 hidden units. After the bidirectional recurrent layers, a fully connected layer is applied as the last layer of the whole model. The output corresponds to a softmax function which computes a probability distribution over characters. This distribution is computed over each timestep. The size of this output layer was equal to the total number of the characters to predict. During the training process, the CTC loss function is computed to measure the error of the predictions, whilst the gradient is computed using backpropagation through time algorithm with the aim of updating the network parameters. The optimizer is the Stochastic Gradient Descent (SGD).

In addition, external language models (LMs) were integrated during the decoding of the E2E systems with the aim of overcoming the initial results. To this end, modified Kneser-Ney smoothed n-grams models of several orders were estimated using the KenLM toolkit [20].

3. Corpora description

In this section, the acoustic and text data used to train and evaluate both E2E and LSTM-HMM systems are presented for each language.

3.1. English

The freely available corpus LibriSpeech [21], a read speech corpus based on audio-books from LibriVox1, was used as dataset. The training, development and testing subsets were maintained as original. These partitions are detailed in Table 1.

Table 1: Training, development and test subsets for English.

<table>
<thead>
<tr>
<th>subset</th>
<th>hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>train-clean</td>
<td>464.2</td>
</tr>
<tr>
<td>train-other</td>
<td>496.7</td>
</tr>
<tr>
<td>dev-clean</td>
<td>5.4</td>
</tr>
<tr>
<td>test-clean</td>
<td>5.4</td>
</tr>
<tr>
<td>dev-other</td>
<td>5.3</td>
</tr>
<tr>
<td>test-other</td>
<td>4.1</td>
</tr>
<tr>
<td>test-noisy</td>
<td>5.4</td>
</tr>
</tbody>
</table>

The clean partitions correspond to those pools with a lower WER in the whole corpus, whilst the other ones contain the most difficult audios a priori. Test-noisy was artificially created within this work using synthesis by superposition [22] of noise samples to the test-clean subset. The noise samples correspond to audios from different acoustic environments selected from the Youtube platform.

Regarding the text data used to train the LMs, they were composed by 22,000 books from Project Gutenberg2 repository, totting up 803 million tokens and 900,000 unique words.

3.2. Spanish

The Spanish subset of the SAVAS corpus [23] was used as the main dataset. It is composed of broadcast news contents from the Basque Country’s public broadcast corporation EITB (Euskal Irrati Telebista), and includes audios in both clear (studio) and noisy (outside) conditions. This media dataset was then transferred through both land- and mobile-lines using different combinations, generating new telephone domain subsets, as it is summarized in Table 2.

Table 2: Training, development and test subsets for Spanish.

<table>
<thead>
<tr>
<th>subset</th>
<th>hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>train-media</td>
<td>132.5</td>
</tr>
<tr>
<td>train-land-mobile</td>
<td>397.5</td>
</tr>
<tr>
<td>dev-media</td>
<td>4</td>
</tr>
<tr>
<td>test-media-clean</td>
<td>4</td>
</tr>
<tr>
<td>test-media-noisy</td>
<td>4</td>
</tr>
<tr>
<td>dev-land</td>
<td>4</td>
</tr>
<tr>
<td>test-land</td>
<td>4.6</td>
</tr>
<tr>
<td>dev-mobile</td>
<td>4</td>
</tr>
<tr>
<td>test-mobile</td>
<td>4.6</td>
</tr>
</tbody>
</table>

Concerning text data, they were obtained by merging transcriptions of the training audios and generic domain news crawled from the Internet. The number of texts summed up a total of 320 million words.

3.3. Basque

The Basque training data was also composed of the Basque subset of the SAVAS corpus, and the audios were gathered from Basque broadcast news programs as well. No telephone domain partition was generated in this case. Table 3 describes the main characteristics of the SAVAS Basque corpus.

The text data were also obtained by merging transcriptions and generic news crawled from the Internet. In total, 180 million words were employed for the LMs estimation.

---

1https://librivox.org/
2https://www.gutenberg.org/
Table 3: Training, development and test subsets for Basque.

<table>
<thead>
<tr>
<th>subset</th>
<th>hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>train-media</td>
<td>142.25</td>
</tr>
<tr>
<td>dev-media</td>
<td>4</td>
</tr>
<tr>
<td>test-media-clean</td>
<td>4</td>
</tr>
<tr>
<td>test-media-noisy</td>
<td>4</td>
</tr>
</tbody>
</table>

4. Baseline E2E and LSTM-HMM systems

Using the above described corpora, the E2E and LSTM-HMM based baseline systems were trained in order to be compared to the evolved ASR systems presented in Section 5. The characteristics of these baseline systems are described in the following subsections.

4.1. Baseline E2E models

Each E2E model per language was trained over the corresponding corpus described above, through a default setup and without applying any enhancement technique. Linear-scale based spectrograms were employed as input. English E2E baseline models were trained using the train-clean and train-other partitions for 15 epochs and a batch size of 10, whilst the Spanish and Basque train-media baselines were built with the train-media partition for 25 epochs and a batch size of 20 because of the corpus characteristics.

4.2. LSTM-HMM models

These models were estimated with the Kaldi toolkit [24]. The acoustic models corresponded to a hybrid LSTM-HMM implementation, where unidirectional LSTMs were trained to provide posterior probability estimates for the HMM states. Besides, modified Kneser-Ney smoothed 3-gram and 5-gram models were used for decoding and re-scoring of the lattices respectively. Both LMs were estimated with the KenLM toolkit.

5. Experiments

5.1. Linear- and Mel-scale based spectrograms

The original Deep Speech 2 architecture employs linear spectrograms as the main audio parametrization method. This experiment was focused on analyzing the use of Mel-scale based spectrograms as input data to the E2E model as it was employed in [25], as it aims to highlight the most relevant information for the human hearing according to the Mel scale.

Two models were compared for the three languages under evaluation; the above presented baseline E2E models trained using linear spectrograms, and the evolved new models estimated with the same configuration and training-set but using Mel-based spectrograms for audio parametrization.

The results in Table 4 show a noticeable improvement for all the experiments except one when using Mel-scale based spectrograms. Focusing on the results without using LM (No LM), a relative improvement of 3% was reached for English on the clean test set, whilst these enhancements were of 8.7% and 10% for Spanish and Basque. This positive behavior was also maintained over the noisy test set, obtaining relative improvements of 1.4%, 7.0% and 6.7% for each corresponding language. The results using LM followed the same tendency, with remarkable differences obtained specially for Basque over the test-media-clean set, where a real improvement of 4 percentage points was reached (from 12.9 to 8.9).

Table 4: WER (%) results for linear- and Mel-based systems for each language.

<table>
<thead>
<tr>
<th>subset</th>
<th>Test-media-clean No LM 3-gram</th>
<th>Test-media-noisy No LM 3-gram</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>10.6 5.6</td>
<td>35.3 23.6</td>
</tr>
<tr>
<td>Mel-scale-EN</td>
<td>10.2 5.4</td>
<td>34.8 22.2</td>
</tr>
<tr>
<td>Spanish</td>
<td>24.0 10.3</td>
<td>39.5 19.2</td>
</tr>
<tr>
<td>Mel-scale-ES</td>
<td>21.9 10.3</td>
<td>36.7 18.9</td>
</tr>
<tr>
<td>Basque</td>
<td>23.8 12.9</td>
<td>38.8 17.3</td>
</tr>
<tr>
<td>Mel-scale-EU</td>
<td>21.2 8.9</td>
<td>36.2 19.2</td>
</tr>
</tbody>
</table>

5.2. Training a mixed model for media and telephone

Telephone speech sampling is usually limited by the bandwidth at which the audio is transmitted through the telephone channel, getting values as much as 8 kHz (4 kHz Nyquist). This commonly implies to train separate acoustic models per domain.

In this work, a mixed E2E model for the media and telephone domains was built, and then a fine-tuning technique was applied over each domain training set to generate domain adapted individual models. It was only performed for the Spanish language and the performance of the resulting systems was compared to the Spanish baseline system. Following the results obtained in the previous experiment, and with the aim of loosing the less valuable telephone speech information as possible, Mel-scale based spectrograms were employed for audio parametrization. In addition, the Spanish train-media dataset was 3-fold augmented through speed perturbation to balance the quantity of data in both domains.

The results achieved over the media test set and the telephone test set are presented in Tables 5 and 6 respectively.

Table 5: WER results for the baseline, mixed (Mixed-ES) and fine-tuned (Mixed-FT-Media and Mixed-FT-Phone) models over the Spanish media test set.

<table>
<thead>
<tr>
<th>subset</th>
<th>Test-media-clean No LM 3-gram</th>
<th>Test-media-noisy No LM 3-gram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline-ES</td>
<td>24.0 10.3</td>
<td>39.5 19.2</td>
</tr>
<tr>
<td>Mixed-ES</td>
<td>16.5 8.8</td>
<td>21.3 11.5</td>
</tr>
<tr>
<td>Mixed-FT-Media</td>
<td>15.7 8.5</td>
<td>20.5 10.9</td>
</tr>
<tr>
<td>Mixed-FT-Phone</td>
<td>18.6 9.6</td>
<td>22.4 11.5</td>
</tr>
</tbody>
</table>

Table 6: WER results for the baseline, mixed (Mixed-ES) and fine-tuned (Mixed-FT-Media and Mixed-FT-Phone) models over the Spanish telephone test set.

<table>
<thead>
<tr>
<th>subset</th>
<th>Test-mobile No LM 3-gram</th>
<th>Test-land No LM 3-gram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline-ES</td>
<td>59.5 33.8</td>
<td>51.7 25.9</td>
</tr>
<tr>
<td>Mixed-ES</td>
<td>22.4 12.8</td>
<td>18.4 9.9</td>
</tr>
<tr>
<td>Mixed-FT-Media</td>
<td>23.4 13.0</td>
<td>18.7 10.0</td>
</tr>
<tr>
<td>Mixed-FT-Phone</td>
<td>22.2 12.1</td>
<td>17.2 9.3</td>
</tr>
</tbody>
</table>
Looking at the results obtained for the media test set in clean conditions, relative improvements of 7.5% and 8.3% of the Mixed-ES and the fine-tuned Mixed-FT-Media models can be observed with respect to the baseline. Besides, improvements of 18.3% were achieved by the fine-tuned Mixed-FT-Media model for the media noisy test set when comparing to the baseline system.

Regarding the results on the telephone domain presented in Table 6, the three models under evaluation present better performance than the Baseline-ES model. As it was expected, the Mixed-FT-Phone model shows the best results with a real improvement of 37.3% with respect to the baseline over the Test-mobile set, composed of audios transferred by the mobile channel. This improvement achieves a 34.5% in the case of the Test-land set. The Mixed-ES model also shows notable enhancements, with slightly worse WER than the Mixed-FT-Phone model. The Mixed-FT-Media performs satisfactorily as well, but it shows a little lower performance than the others.

Finally, it can be also observed that the Mixed-ES model performs almost as well as the fine-tuned models even without having applied any fine-tuning technique.

5.3. Best E2E models compared to LSTM-HMM systems

The aim of this task was to select the best E2E models presented above and compare them to (1) the state-of-the-art models in the literature if available and (2) LSTM-HMM based systems developed on top of the Kaldi toolkit. For this experiment, the selected models for each language were called as Evolved-EN, Evolved-ES and Evolved-EU.

• Evolved-EN. It corresponded to the English baseline model evolved through fine-tuning and a speed-based data augmentation techniques applied for 10 more epochs. The decoding was performed using an external 3-gram LM and using a beam size of 1000.

• Evolved-ES. This model refers to the Spanish fine-tuned Mixed-FT-Media. It was decoded with two configurations; (1) using an external 3-gram LM and a beam size of 600 and (2) with a 5-gram LM with a beam size of 1000.

• Evolved-EU. The selected Basque model was the Mel-scale-EU model presented in subsection 5.1. It was decoded using external 3-gram and 5-gram LM models with a beam size of 600 and 1000 respectively.

In Table 7, a comparison between the evolved model, the Kaldi based LSTM-HMM model, and the results obtained from reference systems in the literature are presented for the English language. Since Test-noisy was created for this work, the results for this subset are only given for the first two models.

Table 8: WER comparison of Kaldi vs E2E for SAVAS Spanish.

<table>
<thead>
<tr>
<th></th>
<th>Test-clean 3-gram</th>
<th>Test-other 3-gram</th>
<th>Test-noisy 3-gram</th>
<th>Test-clean 5-gram</th>
<th>Test-other 5-gram</th>
<th>Test-noisy 5-gram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanish</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evolved-ES</td>
<td>8.5</td>
<td>7.2</td>
<td>10.9</td>
<td>9.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTM-HMM</td>
<td>7.9</td>
<td>7.7</td>
<td>11.9</td>
<td>10.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basque</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evolved-EU</td>
<td>8.9</td>
<td>6.6</td>
<td>19.2</td>
<td>15.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTM-HMM</td>
<td>7.8</td>
<td>6.0</td>
<td>10.8</td>
<td>8.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As it can be observed in Table 7, the evolved E2E system outperformed the LSTM-HMM system and all the reference systems over the clean test. It shows the effect of applying Mel-scale based parametrization, and techniques like fine-tuning and speed-based data augmentation. It has to be also remarked that the WER obtained is 0.9 percentage points lower than the achieved by a human manual transcription.

On the contrary, for the Test-other partition, the LSTM-HMM model obtains a better performance than Deep Speech 1, but still, an error of 2.2 points higher than Deep Speech 2. It has to be considered that the model from Deep Speech 2 was trained with 12,000 hours (including the Librispeech corpus), more than ten times more than the Evolved-EN system (960h).

Table 8 shows how the best Spanish E2E model overcame all the results obtained by the LSTM-HMM based system with the exception of the Test clean when a 3-gram was applied. It shows the robustness of the fine-tuned mixed model trained with data from different acoustic domains. Finally, the scarcity of training data for Basque benefited the LSTM-HMM model against the E2E model, which was estimated using Mel spectrograms as the only enhancement technique.

6. Conclusions and Future Work

In this work, E2E ASR systems for English, Spanish and Basque have been developed and evaluated against reference and LSTM-HMM architectures. Besides, the positive impact of applying some enhancement techniques have been demonstrated through different experimental evaluations.

As main conclusions, it can be stated that using Mel-scale based spectrograms overcomes linear-based ones, as it was proven in all the experiments. Moreover, it was shown that robust hybrid E2E models that perform almost as well as in-domain models can be generated for acoustically different environments. With regard to the training data, as in the case of Basque compared to English, it was clear that more data leads to better results. However, when the resources are limited, using an external LM of a high order can improve the performance, as it was shown for all the languages under evaluation.

As a future work, one important task will be the generation of new Spanish and Basque E2E models with more training data, since these models were still weak without LM. Moreover, current N-grams will be replaced by RNN based LMs, specially for Basque as an agglutinative language. Finally, following novel studies, new research efforts will be made to develop methodologies for a semi-supervised learning within E2E architectures.
7. References


Listening to Laryngectomees: A study of Intelligibility and Self-reported Listening Effort of Spanish Oesophageal Speech.

Sneha Raman, Inma Hernaez, Eva Navas, Luis Serrano

AHOLAB Signal Processing Laboratory, University of the Basque Country (UPV/EHU), Spain
sneha.raman@ehu.eus, inma.hernaez@ehu.eus, eva.navas@ehu.eus, lserrano@aholab.ehu.eus

Abstract

Oesophageal speakers face a multitude of challenges, such as difficulty in basic everyday communication and inability to interact with digital voice assistants. We aim to quantify the difficulty involved in understanding oesophageal speech (in human-human and human-machine interactions) by measuring intelligibility and listening effort. We conducted a web-based listening test to collect these metrics. Participants were asked to transcribe and then rate the sentences for listening effort on a 5-point Likert scale. Intelligibility, calculated as Word Error Rate (WER), showed significant correlation with user rated effort. Speaker type (healthy or oesophageal) had a major effect on intelligibility and effort. Listeners familiar with oesophageal speech did not have any advantage over non familiar listeners in correctly understanding oesophageal speech. However, they reported lesser effort in listening to oesophageal speech compared to non familiar listeners. Additionally, we calculated speaker-wise mean WERs and they were significantly lower when compared to an automatic speech recognition system.

Index Terms: Spoken language understanding, Speech intelligibility, Speech and voice disorders, Pathological speech and language, Speech perception

1. Introduction

Oesophageal speech is an equipment-free speech production method used by people whose larynx has been surgically removed (laryngectomees). In spite of the absence of vocal folds, they can still utter intelligible speech using alternative vibrating elements. In the production of oesophageal speech, the pharyngo-oesophageal segment is used as a substitutive vibrating element. Air is swallowed from the mouth and is introduced into the oesophagus, after which it is expelled in a controlled way, thereby producing vibration. This generation mechanism introduces acoustic artefacts and makes oesophageal speech difficult and effortful to understand [1] [2], which greatly affects communication, interpersonal relationships and hence, quality of life. Moreover, these less intelligible voices are problematic for Automatic Speech Recognition (ASR) systems that are becoming ubiquitous in human-computer interaction technologies. The aim of the work presented in this paper is to quantify the difficulty in understanding oesophageal speech by measuring intelligibility and listening effort. Intelligibility is quantified in both Human Speech Recognition (HSR) and ASR contexts.

Several studies have devised systems for the analysis of pathological speech which also includes intelligibility measurements [3] [4] [5]. In [3] the authors use a Hidden Markov Model based Automatic Speech Recognition (ASR) system to measure objective intelligibility of laryngectomees, and also cleft lip and palate speech. A similar tool for Dutch pathological speech intelligibility calculation is proposed in [5]. In [4] some intelligibility measurement techniques that do not use ASR are described. The main advantage of measures based on ASR is that it is an objective measure and therefore easy to replicate and to implement. However it only evaluates machine intelligibility, and not how intelligible it is to humans. It also does not consider other important factors like pleasantness, acceptability or listening effort.

Some studies have been conducted in measuring the intelligibility of Spanish oesophageal speech. In [6] the voice intelligibility characteristics for Spanish oesophageal and tracheoesophageal speech is reported. This study was conducted for two syllable words. Another study [7] showed how the formant frequencies were higher and the duration of vowels was longer for laryngectomees as compared to healthy speech. The work in [8] describes a real time recognition system for vowel segments of Spanish oesophageal voice.

The above mentioned studies focus on the micro level of words and vowels. Sentence level HSR studies on the intelligibility of Spanish oesophageal voice is a less traversed area of investigation. In this study, sentences are used as our stimuli.

The downside of intelligibility measurements is that they indicate only how many words have been correctly identified but not how difficult it was to identify them. This does not do justice to the problem of how effortful the listening was, especially in adverse listening conditions such as pathological speech. A review of listening effort and various methods of measuring listening effort is presented in [9].

Some research focused on measuring the processing load associated with oesophageal speech. In [10] the authors measured the acceptability of oesophageal, electro-laryngeal and healthy speech. They found that healthy speech was the most acceptable, followed by superior oesophageal speech and then artificial larynx speech. In [11] high intelligibility tracheoesophageal speech was played to listeners and they were asked to rate the effort of listening as well as acceptability for each sample and found an inverse correlation between listening effort and acceptability. Another observation from this study was that even highly intelligible speech can have varying listener effort. In this study we attempt to explore this processing load phenomenon in addition to the ASR and HSR intelligibility measurements. Firstly, we investigate whether the intelligibility (both ASR and HSR) of healthy and disordered speech is comparable. Secondly, we are interested in seeing if the intelligibility and listening effort are correlated.

In [12], the idea of intelligibility differences between experienced and inexperienced listeners of oesophageal speech was explored and the findings stated that oesophageal speech was ranked similarly for intelligibility by both experienced and inexperienced listeners. Following this thread, we were interested in investigating if the same result was observed for our dataset for listeners that are familiar and unfamiliar with oesophageal speech. We consider friends, family and close relatives of oesophageal speakers as familiar listeners.
In short, the hypotheses of the experiment described in this paper are:

- Intelligibility or Word Error Rate measurement is correlated with user rated listening effort.
- Healthy voices are more intelligible and less effortful, compared to oesophageal voices.
- Listeners familiar with oesophageal speech find it less effortful to process, compared to listeners that are not.
- ASR performs worse than HSR for healthy voices, but listeners familiar with oesophageal speech find it less difficult to guess, often containing proper names, dates etc.

We begin by describing the methodology, corpus and details of the listening test. This will be followed by analysis methods used and results. Finally, the conclusions and future work are presented.

2. Methodology

2.1. Experimental Design

The main task for this experiment was the word recall and transcription task: Participants listened to a sentence and then wrote what they have understood. According to [13] the strengths of sentence repetition tasks are that they are "fairly simple cognitive tasks" and that they are "consistent throughout the age span" in the area of neurophysiological tests. Moreover, sentence transcription tasks have been widely used for subjective intelligibility measurements. The work in [14] reports the agreement of sentence transcription tasks with a wide range of intelligibility quantification techniques and in [15] the method is described as "human speech recognition". Therefore, we chose this approach to calculate WER and consequently the intelligibility.

We were also interested in knowing the listening effort of these utterances. We had the participants rate the sentences for listening effort on a 5 point Likert scale. The options were 'very little', 'a little', 'some', 'quite' and 'a lot'.

To avoid priming and sentence order bias, the sentences were played only once and in a random order.

2.2. Corpus and Stimuli

The parallel data used for this task is 100 phonetically balanced sentences selected from a bigger corpus [16], recorded by 35 healthy speakers and 32 oesophageal speakers. The recordings of oesophageal speakers were done in an acoustically isolated room with a studio microphone (Neumann TLM 103). The recordings of the healthy speakers have variable sources because they have been acquired through an online platform [17]. However, some of them were made in the aforementioned acoustically isolated room although with a different microphone.

2.2.1. Selection of Speakers

For this experiment we chose oesophageal speakers based on two criteria: proficiency and accessibility. Proficient speakers were those who underwent laryngectomy, and had begun training to speak for at least two years prior to the recording. Additionally, an oesophageal voice quality assessment tool [18], based on the factors (speaking rate, regularity etc.) of the A4S scale of [19], was used as a guide to assess proficiency. Accessibility of speakers was considered because the opportunity to obtain follow-up recordings could be useful for future research. Based on these criteria, we chose 4 speakers, three male and one female, making it gender inclusive (there are only 4 women in the whole database and only 2 of them fulfilled the two criteria of proficiency and accessibility).

We took the following information from the participants: age, presence of hearing impairment, the kind of audio equipment used (good quality headphones, normal quality head-phones, good loudspeakers and bad equipment) and whether the listener had close contact with laryngectomies.

The listening test was web based and it was possible to 1https://aholab.ehu.eus/users/sneha/Listening_test.php

2.2.2. Selection of Sentences

A pilot listening test was conducted within the lab to assess the feasibility of this corpus for the sentence transcription task. The participants chosen for this pilot study were unfamiliar with the sentences of the corpus and thus not subject to priming. After the pilot test, the participants reported that some sentences were too long to remember and hence, effortful to transcribe. Additionally, although semantically and syntactically correct, the sentences were rich in content and contained words that are difficult to guess, often containing proper names, dates etc.

This led us to reconsider the length of sentences and we decided to choose a subset of shorter sentences, which would make them suitable for sentence transcription. We used the CorpusCRT tool [20] which generates a phonetically balanced subset of sentences based on the provided phonetic criteria. In this case, the criteria we used was a maximum of 40 phonemes and this gave us a set of 30 sentences, each of which had a maximum of 10 words. Some examples of the sentences are the following: "¿Qué diferencia hay entre el caucho y la hevea?" What is the difference between rubber and hevea?, 'Unos días de euforia y meses de atonía.' A few days of euphoria and months of atony.

All the selected sentences (both from oesophageal and healthy speakers) were normalised to a common peak value (0.8) to achieve a homogeneous and comfortable level of loudness.

2.3. The Listening Test

We created six mutually exclusive sets of sentences such that each set contained 30 different sentences and exactly 5 sentences from each speaker. As a result, all 180 sentences (30 sentences from six speakers) was covered after every sixth participant. This ensured equal coverage of all sentences and speakers. Each participant was assigned one of these sets and they listened to the sentences in a random order.

The participants were asked to use headphones for the study unless impossible. They were assured that it was not a test of hearing and that the test was being conducted to obtain their honest and uninhibited response. They were told that the sentence could be played only once and that they should pay close attention and type what they hear. If they missed some portions or were unsure of what they heard, they could put three dots (...) in that place. Additionally, they were asked to mark a response for the amount of effort they experienced for that sample on the aforementioned Likert scale. The first couple of sentences that were presented were practice sentences (one healthy and one oesophageal), to familiarise the participant with the task. These sentences were sampled from the same corpus [21] but different from the ones that appear in the actual test.

We took the following information from the participants: age, presence of hearing impairment, the kind of audio equipment used (good quality headphones, normal quality head-phones, good loudspeakers and bad equipment) and whether the listener had close contact with laryngectomies.

The listening test was web based and it was possible to 1https://aholab.ehu.eus/users/sneha/Listening_test.php
reach out to a wide range of participants. However, this also meant differences in audio equipment and the effects of this on the responses are reported in the results section.

2.4. Automatic Speech Recognition

To have an objective measure of the intelligibility (WER) we prepared an ASR system for Spanish using the Kaldi toolkit [22]. This approach was chosen as it allowed us to control the processing operations followed during the recognition as well as basic aspects of the recognition process such as the lexicon and the language model. It is implemented following the recipe ≤5 for the Wall Street Journal database. The acoustic features used are 13 Mel-Frequency Cepstral Coefficients (MFCCs) to which a process of mean and variance normalization (CMVN) is applied to mitigate the effects of the channel. The details of the training procedures are described in [23].

The audio material used to train the Spanish recogniser was healthy laryngeal speech as described in [24]. However, due to the characteristics of the sentences used for the evaluation, some modifications were made in this ASR system. Although the acoustic models were maintained, a new lexicon was created from the 100 sentences corpus used in the experiment (701 words). This was done because using the original lexicon (with 37,632 entries) as much as 23% of the words were out of vocabulary (OOV) words. This is due to the fact that the sentences are phonetically balanced and many sentences containing proper names and many unusual words were chosen to maximize the variability of the phonetic content. Together with this reduced lexicon, a unigram language model with equally probable words was used.

Although the final WER numbers obtained in this way are not comparable to a realistic ASR situation, the procedure serves our purpose of evaluating the intelligibility of the sentences, comparing the performance of healthy and oesophageal speakers, and establishing a baseline reference for future developments in the field (such as evaluating the improvements of speech modification algorithms).

3. Analysis and Results

We had 57 native Spanish participants in this test, out of which 15 of them had close contact with laryngectomees and hence were familiar with oesophageal speech. The age of listeners ranged from 21 to 70 and mean age was 36.6.

Prior to calculating WER, an initial clean-up was performed on the data. This included removing any punctuations or special characters, and some typing errors (accented vowels, use of upper and lower case, spelling of proper or foreign names etc.). The WER was obtained after correcting these transcription errors.

WER was calculated [25] using the Levenshtein distance between the reference sentence and the hypothesis sentence (the sentence transcribed by the listener). This method calculates the distance by quantifying the insertions, deletions and substitutions that are observed in the hypothesis sentence when compared to the reference sentence.

Self reported listening effort responses were assigned numeric values that ranged from 1 to 5 with 1 corresponding to ‘very little’ and 5 to ‘a lot’.

We performed ANOVA analysis on the dataset using the JASP tool [26] to quantify the effects of speaker type and familiarity of the listener with oesophageal speech. The audio device used by the listener had no effect on the HSR results (F(3,1256)=0.707, p=0.548) and on listening effort (F(3,1256)=0.705, p=0.549).

In addition, we present the WER results from the ASR system for all speakers.

3.1. Word Error Rates from HSR

Table 1 presents mean WERs and Figure 1 shows the speaker-wise WERs for familiar and unfamiliar listeners. OM, OF, HM, HF are acronyms for Oesophageal Male, Oesophageal Female, Healthy Male and Healthy Female respectively. Mean WER is always higher for oesophageal speech compared to healthy speech, as expected. There is no major difference in the WER for familiar and unfamiliar listeners in the case of oesophageal speech. This result corroborates the conclusions in [12]. For healthy speech there is slight difference of around 3 points in the mean WER, but as can be seen in Figure 1 the difference is not meaningful.

The ANOVA results show that familiarity with oesophageal speech had no effect on WER (F(1,1590)=0.360,0.548). On the other hand, speaker-type has a strong effect on WER (F(1,1590)=129.552, p<0.001).

<table>
<thead>
<tr>
<th>Effort</th>
<th>Oesophageal</th>
<th>Healthy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Familiar</td>
<td>2.61</td>
<td>3.07</td>
</tr>
<tr>
<td>Not familiar</td>
<td>3.54</td>
<td>3.265</td>
</tr>
<tr>
<td>Total mean WER</td>
<td>3.07</td>
<td>1.255</td>
</tr>
</tbody>
</table>

WER results (F(3,1256)=0.707, p=0.548) and on listening effort (F(3,1256)=0.705, p=0.549).

3.2. Self-reported Listening Effort

Mean self-reported listening effort values are stated in Table 1 and Figure 2 shows the speaker-wise values. As expected, it is higher for oesophageal speech compared to healthy speech. However, when listening to oesophageal speech the perceived effort is significantly lower for familiar listeners than for not familiar listeners. Indeed, ANOVA analysis shows
that familiarity with oesophageal speech has an effect on effort (F(1,1590)=84.94, p<0.001) and Speaker-type has a strong effect on effort (F(1,1590)=1243.94, p<0.001).

3.3. Correlation of Intelligibility and Listening Effort
Correlation between intelligibility (WER) and self reported effort is 0.479 (Pearson’s r, p <0.001). This is a weak but significant correlation that indicates that sentences with more transcription errors are perceived as more effortful. This relationship between WER and self-reported effort is illustrated as a box-plot in Figure 3.

3.4. Word Error Rates from ASR
The ASR experiment was performed using all the 100 available sentences for each speaker and not only the subset used for human intelligibility measurements. This was convenient in order to obtain a reliable WER measure. It can be observed from Figure 4 that the ASR performs poorly for both healthy and oesophageal speech. The fact that the system is using a unigram language model contributes greatly to this poor performance. As expected, WER for oesophageal speakers is significantly higher than healthy speakers.

Figure 4 also shows the HSR results. We can observe HSR and ASR perform differently for the different speakers. However, the number of speakers in this experiment is small to draw any reliable conclusion about the variation of ASR and HSR across speakers.

4. Conclusions and Future Work
Healthy voices are on an average three times as intelligible as oesophageal voices. The mean self reported effort was also three times larger for oesophageal speech compared to healthy voices. There was significant correlation between intelligibility and effort. Speaker type had an effect on both intelligibility and effort. Listeners familiar with oesophageal fared the same for intelligibility as people who were not. However, they reported less effort in listening to oesophageal speech than the not familiar listeners. The ASR system we chose for this task had poorer WER for oesophageal voice compared to healthy voice.

The listening effort obtained through this study is based on the listener’s own interpretation of ‘effort involved in listening’. This will provide us with a reference for comparison when we perform objective listening effort measurements in the future using physiological methods such as EEG and pupillometry. If these subjective measures are found to be correlated with the physiological measurements, then that opens the possibility of using the less cumbersome self report strategy to achieve our purpose of evaluation.

Both HSR intelligibility and ASR intelligibility play different but important roles in oesophageal speech evaluation. While improved HSR would enable better human-human interactions, an improved ASR performance would enable better human-machine interactions (eg. digital voice assistants). Lower listening effort would also contribute towards better communication with humans.

Our main future work is to build an oesophageal voice restoration system aimed at better ASR and HSR intelligibility and low listening effort.

5. Acknowledgements
This project has received funding from the European Unions H2020 research and innovation programme under the Marie Curie European Training Network ENRICH (675324). The work has been partially funded by the Spanish Ministry of Economy and Competitiveness with FEDER support (MINECO/FEDER, UE) (RESTORE project, TEC2015-67163-C2-1-R).
6. References

[1] B. Weinberg, “Acoustical properties of esophageal and tracheoe-

[2] T. Most, Y. Tobin, and R. C. Minran, “Acoustic and perceptual characteristics of esophageal and tracheoesophageal speech pro-


phonological and acoustic ast-free features for pathological

“Dia: a tool for objective intelligibility assessment of pathological


cal analysis of spanish vowels produced by laryngectomized sub-


dated norms and clinical utility,” in 7, pp. 154–9, 02 2000.


[16] N. Tits, “Exploring the parameters describing the quality and in-
telligibility of alaryngeal voices,” University of Mons, 2017.

[17] T. Drugman, M. Rijckaert, C. Janssens, and M. Remacle, “Tra-
chezoesophageal speech: A dedicated objective acoustic assess-


Towards an Automatic Evaluation of the Prosody of Individuals with Down Syndrome.

Mario Corrales-Astorgano\textsuperscript{1}, Pastora Martínez-Castilla\textsuperscript{2}, David Escudero-Mancebo\textsuperscript{3}, Lourdes Aguilar\textsuperscript{3}, César González-Ferreras\textsuperscript{3}, Valentín Cardeñoso-Payo\textsuperscript{1}

\textsuperscript{1}Department of Computer Science, University of Valladolid, Spain
\textsuperscript{2}Department of Developmental and Educational Psychology, UNED, Spain
\textsuperscript{3}Department of Hispanic Philology, Universitat Autònoma de Barcelona, Spain
mcorrales@infor.uva.es, pastora.martinez@psi.uned.es, descuder@infor.uva.es, lourdes.aguilar@uab.cat, cesargf@infor.uva.es, valen@infor.uva.es

Abstract

Prosodic skills may be powerful to improve the communication of individuals with intellectual and developmental disabilities. Yet, the development of technological resources that consider these skills has received little attention. One reason that explains this gap is the difficulty of including an automatic assessment of prosody that considers the high number of variables and heterogeneity of such individuals. In this work, we propose an approach to predict prosodic quality that will serve as a baseline for future work. A therapist and an expert in prosody judged the prosodic appropriateness of individuals with Down syndrome’s speech samples collected with a video game. The judgments of the expert were used to train an automatic classifier that predicts the quality by using acoustic information extracted from the corpus. The best results were obtained with an SVM classifier, with a classification rate of 79.30%. The difficulty of the task is evidenced by the high inter-human rater disagreement, justified by the speakers’ heterogeneity and the evaluation conditions. Although only 10% of the oral productions judged as correct by the referees were classified as incorrect by the automatic classifier, a specific analysis with bigger corpora and reference recordings of people with typical development is necessary.

Index Terms: Prosody, Automatic Classification, Down syndrome, Educational Video games

1. Introduction

The collective of individuals with Down syndrome shows a series of cognitive, learning and attentional limitations. All the areas of language are altered, but not in the same degree, as described in [1]. Although lexical acquisition is delayed, morphology and syntax appear to be more affected than vocabulary [2]. Related to pragmatics, individuals with Down syndrome show difficulties when producing and understanding questions and emotions, signaling turn-taking, or keeping topics in conversation, and the study in [3] demonstrated that children with Down syndrome are impaired relative to norms from typically developing children in all areas of pragmatics. At phonological level, speech intelligibility is seriously damaged by the presence of errors on producing some phonemes, the loss of consonants and the simplification of syllables [4].

What concerns to prosody, [5] report disfluencies (stuttering and cluttering) and impairments in the perception, imitation and spontaneous production of prosodic features; authors of [6] have connected some of the speech errors with difficulties in the identification of boundaries between words and sentences. Nevertheless, characterizing prosodic impairments in populations with developmental disorders is a hard task [7]. To fulfill such an aim, prosody assessment procedures appropriate for use with individuals with intellectual and/or developmental disabilities need to be employed. The Profiling Elements of Prosody in Speech-Communication (PEPS-C) test has proved to be successful in this respect [8, 9]. When used with English-speaking children with Down syndrome, lower performance than expected by chronological age is observed in all prosody tasks [10]. After comparisons with typically developing children matched for mental age, impairments are also found for the discrimination and imitation of prosody [10].

There are technological tools focused on language therapy [11, 12]. However, the difficulty of separating the effects of each of the suprasegmental features on communication together with the multiplicity of right possibilities to arrive to the same intonational meaning explains that little attention has been paid to the development of technological resources that specifically consider the learning of prosody in students with special needs, specifically in those with Down syndrome. To advance in the line of developing specific resources to minimize the limitations concerning prosody and pragmatics in individuals with Down syndrome, we have developed an educational video game to train prosody, PRADIA: Mistery in the city [13, 14].

Although the video game was designed with the aim of training prosody in individuals with Down syndrome, it became a tool to collect their oral productions and thus to construct a prosodic corpus. These aims are achieved thanks to the fact that the main way of interaction of the player with the game is through the voice. To advance in the game, the player must give an adequate oral response in different communicative circumstances, where prosodic features are the most relevant to achieve a correct pragmatic interpretation. In its current version, the video game needs the constant presence of a person (ideally a therapist) who guides the gamer throughout the adventure and who evaluates the success in the resolution of the production activities. The assistance of the therapist has been proved crucial to motivate individuals with Down syndrome. Even so, it would be desirable to improve their autonomy and to help trainers in their therapies with new functionalities by including a module of automatic assessment of prosodic quality.

If we turn to the field of automatic assessment, the attempts...
of classifying different speech dimensions is well researched, but focused on specific aspects or reduced populations. Some works focus on speech intelligibility of people with aphasia [15] or speech intelligibility in general [16]. Others try to identify speech disorders in children with cleft lip and palate [17]. In addition, speech emotions and autism spectrum disorders recognition have been investigated [18]. The point is that all these works include a subjective evaluation done by experts as a gold standard to train the classification systems.

In this work, we analyse the difficulties of automatically predicting the quality of the prosody of an oral production and propose a new approach that will serve as a baseline for future work. Recordings of individuals with Down syndrome collected in different sessions of use of the educational video game PRA-DIA: Mystery in the city were used to obtain information about the relevant features needed to make an automatic classification of the productions. The speech corpus obtained along the time of game was judged by a therapist, who evaluated in real time the quality of the oral productions, and by a prosody expert, who did an off-line evaluation. The difference in the experimental procedure will be used to investigate if an automatic system can only rely on prosodic variables to judge the oral productions of the players (offline evaluation), or whether other features related to the game dynamics should also be incorporated in the system. The judgments of the expert are used to train an automatic classifier that predicts quality by using acoustic information extracted from the audios of the corpus.

In section 2, the experimental procedure is described, which includes the procedure for corpus collection, the processing of speech material and the classification of the samples. The results section shows the effectiveness of the procedure, although, at the same time, the difference in the evaluation procedures highlights the need of carefully defining the selection of features in the process of classification. We end the paper with a discussion about the relevance of the results and the conclusions and future work section.

2. Experimental procedure

2.1. Corpus description

The three subcorpora were recorded using the video game, but the version of the video game and the recording context were different. The complete description of each subcorpus can be seen on Table 1.

To build the subcorpus C1, five young adults with DS (mean age 198 months) were recruited from a local Down syndrome Foundation located in Madrid (Spain). To account for the variability often found in individuals with Down syndrome and get measurements of different developmental variables, all of the participants were administered with the following tests. The Peabody Picture Vocabulary Scale-III [19] was used to assess verbal mental age, the forward digit-span subtest included in the Wechsler Intelligence Scale for Children-IV [20] was employed to evaluate verbal short-term memory and Raven’s Coloured Progressive Matrices [21] served as a means to measure non-verbal cognitive level. Descriptive characteristics and scores obtained are shown in Table 2. The full PEPS-C battery in its Spanish version [22] was also administered to participants to have specific measurements of prosody level. Mean percentage of success in perception and production PEPS-C tasks is also presented in Table 2. Once these assessments were completed, participants were administered with the PRADIA video game. Each participant used PRADIA for a total duration of 4 hours, distributed in 4 sessions of 1 hour per week. Participants were supported by a speech and language therapist who knew them in advance and was an expert at working with individuals with Down syndrome. The therapist explained the game, helped participants when needed, and took notes about how each session developed. Importantly, the therapist also assessed participants’ speech productions and thus monitored their rhythm of progress within the video game.

C2 subcorpus was also recorded using PRADIA software. These recordings were obtained through the video game within one session of software testing with real users. This test session was done in a school of special education located in Valladolid (Spain). Five adults with Down syndrome, aged 18 to 25, participated in this test. The judgments obtained during this game session were discarded for this work because the speech productions were not evaluated by a therapist. The oral productions were judged in an offline mode by the expert in prosody.

C3 subcorpus was recorded using an older version of PRADIA software, the Magic Stone [23], with less types of production activities. Eighteen young adults with Down syndrome participated in the different game sessions, which focused on how these users interacted with the video game. Five of these eighteen speakers participated as well in the recordings of the C2 subcorpus, so their productions were discarded from C3 subcorpus. As well as in the C2 subcorpus, the judgments decided by the assistant that helped players complete the adventure were not considered in the classifications. Instead, the oral productions were judged in an offline mode by the expert in prosody.

2.2. Corpus evaluation

During the game sessions, a speech therapist sits next to the player and evaluates the production activities in real time. Consequently, in C1 corpus, the therapist adapted her judgments to both the general developmental level of participants and their
Table 2: Description of the C1 subcorpus. For each speaker, this table shows Chronological age (CA), Verbal mental age (VA), Short-term verbal memory (STVM), and Non-verbal cognitive level (NVCL). Ages are expressed in months. In addition, the mean percentage of success in perception (MPercT) and production (MProdT) PEPS-C tasks are included.

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Gender</th>
<th>CA</th>
<th>VA</th>
<th>STVM</th>
<th>NVCL</th>
<th>MPercT</th>
<th>MProdT</th>
</tr>
</thead>
<tbody>
<tr>
<td>S01</td>
<td>f</td>
<td>195</td>
<td>94</td>
<td>94</td>
<td>10</td>
<td>69.70%</td>
<td>45.98%</td>
</tr>
<tr>
<td>S02</td>
<td>m</td>
<td>204</td>
<td>99</td>
<td>134</td>
<td>18</td>
<td>76.04%</td>
<td>72.10%</td>
</tr>
<tr>
<td>S03</td>
<td>f</td>
<td>178</td>
<td>96</td>
<td>78</td>
<td>20</td>
<td>73.96%</td>
<td>74.65%</td>
</tr>
<tr>
<td>S04</td>
<td>m</td>
<td>190</td>
<td>60</td>
<td>below 74</td>
<td>10</td>
<td>60.42%</td>
<td>49.76%</td>
</tr>
<tr>
<td>S05</td>
<td>m</td>
<td>223</td>
<td>69</td>
<td>below 74</td>
<td>13</td>
<td>56.25%</td>
<td>54.84%</td>
</tr>
</tbody>
</table>

2.4. Automatic classification

As explained in section 2.2, the recordings were evaluated by the therapist and the prosody expert. Since the final aim of the module is to decide if the gamer can continue the game or should repeat the activity (without considering degrees of failure), the evaluation of the expert was used to build the classifier. According to this, the output of the different classifiers are Right (R) or Wrong (W), based on the prosody expert scoring. The Weka machine learning toolkit [28] was used and three different classifiers were used to compare their performance: the C4.5 decision tree (DT), the multilayer perceptron (MLP) and the support vector machine (SVM). In addition, the results of using the recordings of the three corpora as well as all combinations of these corpora were compared.

Furthermore, the stratified 10-fold cross-validation technique was used to create the training and testing datasets. We also used feature selection before training the classifiers: the features were selected by measuring the information gain of the features. We used the Weka machine learning toolkit [28] and three different classifiers were used: SVM classifier works better with the classifier used (Table 3). SVM classifier works better with all corpora and the worst results are obtained using DT classifier (best case is 79.3% vs 64.94% baseline). The best results are obtained in Case A and D by using any of the three classifiers (UAR 0.83 with SVM classifier). The classification accuracy decreases when the C3 corpus is entered (C, E, F and G cases) as the number of speakers substantially increases. Moreover, when the same features are used to identify speakers instead of the quality of the productions, with a higher percentage of W assignments from the prosody expert (42.75% and 49% respectively) and higher percentage of Rep. from the therapist (47% and 50%, respectively).

The classification results highly depend on the corpus and the classifier used (Table 3). SVM classifier works better with all corpora and the worst results are obtained using DT classifier (best case is 79.3% vs 64.94% baseline). The best results are obtained in Case A and D by using any of the three classifiers (UAR 0.83 with SVM classifier). The classification accuracy decreases when the C3 corpus is entered (C, E, F and G cases) as the number of speakers substantially increases. Moreover, when the same features are used to identify speakers instead of the quality of the utterance (column #SR rate of Table 3), scenarios Case C, E and G are the worst ones and scenarios Case A and B are the best. In order to see the influence of the speaker in the classification results, we present results per speaker in Table 4.

We focus on Case D to present results per speakers in Table 4. Only the samples of corpus C1 are analyzed because they were evaluated by the two evaluators. Comparing the R-W judgments of the expert with the classifier predictions, there is a high recall in R-R case for all speakers (S01 83.91%, S02 87.65%, S03 97.44%, S04 94.67%, S05 87.01%). The coincidence in W-W case is lower: while S02 and S05 present a reasonable classification rate (72% and 70.27%, respectively), results for S03 goes down to 26.32%. Concerning this result, we note that most of the utterances judged as wrong by the expert were rated as right by the therapist (100% in cell W-Cont.R for S3). As average, we obtain only 10.05% of false negatives. This will be discussed in the next section as a positive result for emotional and motivational level. The video game allows to evaluate the result of the oral activities typing a concrete key on the computer keyboard where the game is installed. If the evaluation is Cont.R (Continue with right result) or Cont (Continue but the oral activity could be better), the video game advances to the next activity. If the evaluation is Rep. (Repeat), the game offers a new attempt in which the player has to repeat the activity. For each activity, there is a predetermined number of attempts: when the attempts finish, the video game goes to the next screen to avoid frustration on the player, even if the activity has not been successfully completed (and the therapist continues judging with Rep.).

On the other hand, an expert in prosody evaluated the three subcorpora of oral productions of 23 speakers with Down syndrome in an offline mode. Due to the difficulty of the task and the different context of the evaluation, the prosody expert used a reduced evaluation system (Right or Wrong production). The judgments were made relying on purely auditive basis, without any acoustic analysis of the sentences, and the focus was on the intonational and prosodic structure. Related to this, factors of intelligibility, quality in pronunciation or adjustment to the expected sentence were not taken into account. Even in the case of speakers with low cognitive level and serious problems of intelligibility, the main criterion was whether they had modeled prosody with certain success, even if the message was not understood. Following the categories of intonational phonology [24] and the learning objectives included in PRADIA [14], criteria concerning intonation, accent and prosodic organization were used to judge if the sentence was Right or Wrong: in short, adjustment to the expected modality; respect for the difference between lexical stress and accent (tonal prominence); and adjustment to the organization in prosodic groups relying mainly in the distinction between function and content words.

2.3. Feature extraction

The openSmile toolkit [25] was used to extract acoustic features from each recording of C1, C2 and C3 subcorpora. The GeMAPS feature set [26] was selected due to the variety of acoustic and prosodic features contained in this set. This set contains frequency related features, energy related features, spectral features and temporal features. The arithmetic mean and the coefficient of variation were calculated on these features. Furthermore, 4 additional temporal features were added: the silence and sounding percentages, silences per second and the mean silences. The complete description of these features can be found in previous research [27]. In this work, only prosodic features (frequency, energy and temporal) have been used because spectral features improve the speaker identification, and classifiers can be adapted to each speaker in the classification process. In total, 34 prosodic features were employed.
Table 3: Classification results depending on the corpus and the classifier used. The prosody expert judgments were used to train the classifier. BL means the performance baseline of each group of samples (number of samples of the most populated class divided by all the samples). DT means Decision trees, SVM means Support vector machines and MLP means Multilayer Perceptron. CR means the classification rate, AUC means the Area Under the Curve and AUR means the Unweighted Average Recall. The number of samples (utt.), the number of speakers (SPK), the number of features (Feat.) and the speaker classification rate using SVM (SR rate) are presented. The lower part of Table 3 shows the different classifiers: Right or Wrong, based on prosody expert scoring.

<table>
<thead>
<tr>
<th>Corpora</th>
<th>BL</th>
<th>CR</th>
<th>SVM</th>
<th>MLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case A</td>
<td>C1</td>
<td>65.79%</td>
<td>69.57%</td>
<td>62.5%</td>
</tr>
<tr>
<td>Case B</td>
<td>C2</td>
<td>61.90%</td>
<td>60.26%</td>
<td>72.68%</td>
</tr>
<tr>
<td>Case C</td>
<td>C3</td>
<td>50.78%</td>
<td>65.76%</td>
<td>61.58%</td>
</tr>
<tr>
<td>Case D</td>
<td>C1+C2</td>
<td>64.94%</td>
<td>70.77%</td>
<td>79.3%</td>
</tr>
<tr>
<td>Case E</td>
<td>C1+C3</td>
<td>62.16%</td>
<td>66.29%</td>
<td>72.31%</td>
</tr>
<tr>
<td>Case F</td>
<td>C2+C3</td>
<td>55.96%</td>
<td>60.94%</td>
<td>66.47%</td>
</tr>
<tr>
<td>Case G</td>
<td>C1+C2+C3</td>
<td>62.11%</td>
<td>66.88%</td>
<td>74.32%</td>
</tr>
</tbody>
</table>

Table 4: Percentage of coincidence between therapist decision, classifier (SVM in case D) and prosody expert per speaker. Concerning the classifier, R represents the utterances classified as Right by the classifier and W represents the utterances classified as Wrong by the classifier. Each row percentage is relative to the number of each type of utterances of prosody expert evaluation.

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Total utt</th>
<th>Type</th>
<th>Total</th>
<th>Expert judgment</th>
<th>Classified as</th>
<th>Therapist decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>S01</td>
<td>120</td>
<td>R</td>
<td>41</td>
<td>84.98% 18.00%</td>
<td>89.90% 24.44% 0.99%</td>
<td></td>
</tr>
<tr>
<td>S02</td>
<td>106</td>
<td>R</td>
<td>31</td>
<td>57.56% 24.22%</td>
<td>55.19% 16.00% 0.00%</td>
<td></td>
</tr>
<tr>
<td>S03</td>
<td>97</td>
<td>R</td>
<td>78</td>
<td>97.44% 2.56%</td>
<td>96.8% 3.8% 1.28%</td>
<td></td>
</tr>
<tr>
<td>S04</td>
<td>111</td>
<td>R</td>
<td>56</td>
<td>92.51% 5.39%</td>
<td>91.55% 5.05% 0.00%</td>
<td></td>
</tr>
<tr>
<td>S05</td>
<td>133</td>
<td>R</td>
<td>74</td>
<td>92.71% 7.29%</td>
<td>92.71% 7.29% 0.00%</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>605</td>
<td>R</td>
<td>350</td>
<td>80.96% 10.05%</td>
<td>80.29% 86.79% 35.46%</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Percentage of coincidence between therapist decision, classifier (SVM in case D) and prosody expert per speaker. Concerning the classifier, R represents the utterances classified as Right by the classifier and W represents the utterances classified as Wrong by the classifier. Each row percentage is relative to the number of each type of utterances of prosody expert evaluation.

real time situations.

Concerning the therapist judgments, Cont. R decision could be identified as a Right assignment in a high percentage of cases for S01, S02 and S03 speakers (69%, 85% and 95% respectively). They are the participants with higher developmental level, according to Table 2. Among these three participants, the first one—with the lowest inter-judge agreement—showed the lowest prosodic level from the outset. In general, the correspondence between real time decisions and expert judgment is not straightforward, with a high variety in the contingency table. Concerning the therapist Rep. decision, it is clear that the highest percentages of agreement are obtained for S04 and S05 speakers (62.5% and 72.97%, respectively), who are the least qualified speakers in Table 2.

4. Discussion and Conclusions

The study shows some of the variables that contribute to account for the difficulties of conducting an automatic evaluation of prosody in Down syndrome. As shown in Table 2, the chronological age of the participants for whom both the therapist and prosody expert evaluations were available was similar. However, their skills for reasoning, recalling auditory verbal material and understanding vocabulary were clearly different. Pheno-
type variability is common in Down syndrome [3] and needs to be considered if prosody is to be evaluated. When developmental level is low, the quality of the prosodic productions is also low. As a result, the likelihood of human agreement as to the appropriateness of the output decreases. This shows the difficulties inherent to the task being carried out. Furthermore, even in the cases of higher cognitive level, variability in the linguistic profile can also play a role. Thus, levels of vocabulary are not necessarily paired with those of prosody perception and production. Differences in the evaluation context also explain the variability between the expert and therapist’s judgments. While the former only based her decisions on intonational criteria, the latter also took into consideration the progress of the player within the video game. In doing so, avoiding frustration was a priority; therefore, levels of frustration tolerance and number of failures influenced the therapist’s decisions.

In our video game, not to evaluate as wrong a right utterance is very important; otherwise, frustration may arise. This is even more important when individuals with Down syndrome are the players since they can be particularly prompted to this type variability is common in Down syndrome [3] and needs to be considered, in the line of what the therapist did in her evaluation. The player profile—among other features related to the progress in the game—should also be incorporated in the system. In addition, the evaluation scale can be improved by adding more dimensions to be scored by the experts. Instead of having a global score of the prosody of a recording, the experts could assign a different score to different prosodic dimensions (intonation, pauses, rhythm), with the aim of making a more precise classification.

The high variability of speech of individuals with Down syndrome has been evidenced in experimental results. Further research should compile a bigger and more balanced corpus of the speech of individuals with Down syndrome and record a reference corpus of people with typical development. Nevertheless, inter-speaker variability should be considered as an intrinsic feature of the voices of individuals with Down syndrome so that both the reference of correctness and the particular limitation of the speaker must be taken into account to attain an effective automatic prosodic assessment.
5. References


Whispered-to-voiced Alaryngeal Speech Conversion with Generative Adversarial Networks

Santiago Pascual1, Antonio Bonafonte1, Joan Serra2, Jose A. Gonzalez3

1Universitat Politècnica de Catalunya, Barcelona, Spain
2Telefónica Research, Barcelona, Spain
3Universidad de Málaga, Málaga, Spain
santi.pascual@upc.edu

Abstract
Most methods of voice restoration for patients suffering from aphonia either produce whispered or monotone speech. Apart from intelligibility, this type of speech lacks expressiveness and naturalness due to the absence of pitch (whispered speech) or artificial generation of it (monotone speech). Existing techniques to restore prosodic information typically combine a vocoder, which parameterises the speech signal, with machine learning techniques that predict prosodic information. In contrast, this paper describes an end-to-end neural approach for estimating a fully voiced speech waveform from whispered alaryngeal speech. By adapting our previous work in speech enhancement with generative adversarial networks, we develop a speaker-dependent model to perform whispered-to-voiced speech conversion. Preliminary qualitative results show effectiveness in re-generating voiced speech, with the creation of realistic pitch contours.

Index Terms: pitch restoration, whispered speech, generative adversarial networks, alaryngeal speech.

1. Introduction
Whispered speech refers to a form of spoken communication in which the vocal folds do not vibrate and, therefore, there is no periodic glottal excitation. This can be intentional (e.g., speaking in whispers), or as a result of disease or trauma (e.g., patients suffering from aphonia after a total laryngectomy). The lack of pitch reduces the expressiveness and naturalness of the voice. Moreover, it can be a serious impediment for speech intelligibility in tonal languages [1] or in the presence of other interfering sources (i.e., cocktail party problem [2]). The conversion from whispered to voiced speech, either by reconstructing partially existent pitch contours or by generating completely new ones, is an area of research that not only has relevant practical and real-world applications, but also fosters the development of advanced speech conversion systems.

In general, existing methods for whispered-to-voiced speech conversion either follow a data-driven or an analysis-by-synthesis approach. In the data-driven approach, machine learning is used to estimate the pitch from the available speech parameters (e.g., mel frequency cepstral coefficients; MFCCs). Then, a vocoder is used to synthesize speech from those by, for instance, predicting fundamental frequencies and voiced/unvoiced decisions from frame-based spectral information of the whispered signal, using Gaussian mixture models (GMMs) [3, 4, 5] or deep neural networks [6]. The analysis-by-synthesis approach follows a similar methodology to code-excited linear prediction [7, 8, 9]. To estimate pitch parameters, a common strategy is to derive those from other parameters available in the whispered signal, such as estimated speech formants [10].

A key application of whisper-to-voiced speech conversion is to provide individuals with aphonia with a more naturally sounding voice. People who have their larynx removed as a treatment for cancer inevitably lose their voice. To speak again, laryngectomies can resort to a number of methods, such as the voice valve, which produces an unnatural, whispered sound, or the electrolarynx, a vibrating device placed against the neck that generates the lost glottal excitation but, nonetheless, produces a robotic voice due to its constant vibration. In recent years, the use of whisper-to-speech reconstruction methods [3, 4, 5, 11], or silent speech interfaces [12, 13], in which an acoustic signal is synthesised from non-audible speech-related biosignals such as the movements of the speech organs, have started to be investigated to provide laryngectomies with a better and more naturally sounding voice.

In this paper, and in contrast to previous approaches, we present a speaker-dependent end-to-end model for voiced speech generation based on generative adversarial networks (GANs) [14]. With an end-to-end model directly performing the conversion between waveforms, we avoid the explicit extraction of spectral information, the error-prone prediction of intermediate parameters like pitch, and the use of a vocoder to synthesize speech from such intermediate parameters. With a GAN learning the mapping from whispered to voiced speech, we avoid the specification of an error loss over raw audio and make our model truly generative, thus being able to produce new realistic pitch contours. Our results show this novel pitch generation as an implicit process of waveform restoration. To evaluate our proposal, we compared the pitch contour distributions predicted by our proposal with those obtained by a regression model, observing that our proposal is able to attain more natural pitch contours than those predicted by a regression model, including a more realistic variance factor that relates to more expressiveness.

The remainder of the paper is structured as follows. In section 2 we describe the method used to restore the voiced speech with GANs. The experimental setup is described in section 3, which includes descriptions of the dataset, an RNN baseline and the hyper-parameters for our GAN model. Finally, sections 4 and 5 contain the discussions of results and conclusions respectively.

2. Generative Adversarial Networks for Voiced Speech Restoration
The proposed model is an improvement over our previous work on speech enhancement using GANs (SEGAN) [15, 16] in order
to handle speech reconstruction tasks. SEGAN was designed as a speaker- and noise-agnostic model to generate clean/enhanced versions of aligned noisy speech signals. From now on, we change signal names for the new task, so we rather work with natural, voiced (i.e. restored) and whispered speech signals. To adapt the architecture to the task of voiced speech restoration, we decide to remove the audio alignment requirement, as the data we use has slight misalignments between input and output speech (see Section 3.1 for more details). In addition, we introduce a number of improvements that consistently stabilize and facilitate its training after direct regularization over the waveform is removed. These modifications also refine the generated quality at the generator output when regression is removed.

2.1. SEGAN

We now outline the most basic aspects of SEGAN, specifically highlighting the ones that have been subject to change. For the sake of brevity we refer the reader to the original paper and code [15] for more detailed explanations on the old architecture and setup. The SEGAN generator network (G) embeds an input noisy waveform chunk into the latent space via a convolutional encoder. Then, the reconstruction is made in the decoder by ‘deconvolving’ back the latent signals into the time domain. G features skip connections with constant factors (acting as identity functions) to promote that low-level features could escape a potentially unnecessary compression from the encoder. Such skip connections also improve training stability, as they allow gradients to flow better across the deep structure of G, which has a total of 22 layers. In the denoising setup, an L1 regularization term helped centering output predictions around 0, discouraging G to explore bizarre amplitude magnitudes that could make the discriminator network (D) converge to easy discriminative solutions for the fake adversarial case.

2.2. Adapted SEGAN

The SEGAN architecture has been adapted to cope with misalignments in the input/output signals as mentioned before, as well as to achieve a more stable architecture and to produce better quality outputs. In the current setup, similarly to the original SEGAN mechanism, we inject whisperer data to G, which compresses it and then recovers a version of the utterance with prosodic information.

The spectral regularization is added to the adversarial loss in the input/output signals as mentioned before, as well as to achieve a more stable architecture and to produce better quality outputs. In the current setup, similarly to the original SEGAN mechanism, we inject whisperer data to G, which compresses it and then recovers a version of the utterance with prosodic information. To cope with misalignments, we get rid of the L1 regularization term, as this was forcing a one-to-one correspondence between audio samples, assuming input and output had the same phase. In its place we use a softer regularization which works in the spectral domain, similar to the one used in the parallel Wavnet [17]. We use a non-averaged version of this loss though, as we work with large frames during training (16,384 samples per sequence), and averaging the spectral frames over this large span could be ineffective. Moreover, we calculate the loss as an absolute distance in decibels over, we calculate the loss as an absolute distance in decibels.

The SEGAN architecture has been adapted to cope with misalignments in the input/output signals as mentioned before, as well as to achieve a more stable architecture and to produce better quality outputs. In the current setup, similarly to the original SEGAN mechanism, we inject whisperer data to G, which compresses it and then recovers a version of the utterance with prosodic information. To cope with misalignments, we get rid of the L1 regularization term, as this was forcing a one-to-one correspondence between audio samples, assuming input and output had the same phase. In its place we use a softer regularization which works in the spectral domain, similar to the one used in the parallel Wavnet [17]. We use a non-averaged version of this loss though, as we work with large frames during training (16,384 samples per sequence), and averaging the spectral frames over this large span could be ineffective. Moreover, we calculate the loss as an absolute distance in decibels.

The spectral regularization is added to the adversarial loss coming from D with a weighting factor λ. In SEGAN, D is a learnable comparative loss function between natural or voiced signals and whispered ones. This means we have a (natural, whispered) paired input as a real batch sample and (voiced, whispered) as a fake batch sample. In contrast, G has to make (voiced, whispered) true, thus being the adversarial objective. In the current setup, we add an additional fake signal in D that will enforce the preservation of intelligibility when we forward data through G: the (natural, random_natural_shuffle) pair. This pair tries to send messages to G about a bad behavior whenever the content between both chunks, the one coming from G and the reference one, changes. Note that we are using the least-squares GAN form (LSGAN) in the adversarial component, so our loss functions, for D and G respectively, become

$$ \min_D V(D) = \frac{1}{3} E_{x, \tilde{w} \sim p_{raw}(\cdot)}[(D(x, \tilde{w}) - 1)^2] + \frac{1}{3} E_{z \sim p_{z}(\cdot)}[D(G(z, \tilde{w}), \tilde{w})^2] + \frac{1}{3} E_{x' \sim p_{data}(\cdot)}[D(x, x')^2] $$

$$ \min_G V(G) = E_{x' \sim p_{data}(\cdot)}[D(G(z, \tilde{w}), \tilde{w}) - 1]^2, $$

where \( \tilde{w} \in \mathbb{R}^T \) is the whispered utterance, \( x \in \mathbb{R}^T \) is the natural speech, \( x' \in \mathbb{R}^T \) is a randomly chosen natural chunk within the batch, \( G(z, \tilde{w}) \in \mathbb{R}^T \) is the enhanced speech, and \( D(x, \tilde{w}), D(G(z, \tilde{w}), \tilde{w}), D(x, x') \) are the discriminator decisions for each input pair. All of these signals are vectors of length T samples except for D outputs, which are scalars. T is a hyper-parameter fixed during training but it is variable during test inference.

After removing the regularization factor L1, the generator output can explore large amplitudes whilst adapting to mimic the speech distribution. As a matter of fact, this collapsed the training whenever the tanh activation was placed in the output layer of G to bound its output to \([-1, 1]\), because the amplitude grew quickly with aggressive gradient updates and tanh would not allow G to properly update anymore due to saturation. The way to correct this was bounding the gradient of D by applying spectral normalization as proposed in [18]. The discriminator does not have any batch normalization technique in this implementation, and its architecture is the same as in our previous work.
The new design of $G$ is shown in Figure 1. It remains as a fully convolutional encoder-decoder structure with skip connections, but with two changes. First, we reduce the number of layers by augmenting the pooling factor from 2 to 4 at every encoder-decoder layer. This is in line with preliminary experiments on the denoising task, where increasing pooling has been effective to improve objective scores for that task. Second, we introduce learnable skip connections, and these are now summed instead of concatenated to decoder feature maps. We thus have now learnable vectors $a_l$, which multiply every channel of its corresponding shuttle layer $l$ by a scalar factor $\alpha_{l,k}$. These factors are all initialized to one. Hence, at the $j$-th decoder layer input we have the addition of the $l$-th encoder layer response following

$$h_j = h_{j-1} + a_l \odot h_l,$$

where $\odot$ is an element-wise product along channels.

### 3. Experimental Setup

To evaluate the performance of our technique, a clinical application involving the generation of audible speech from captured movement of the speech articulators is tested. More details about the experimental setup in terms of dataset, baseline and hyper-parameters for our proposed approach are given below.

#### 3.1. Task and Dataset

In our previous work [6, 13], a silent speech system aimed at helping laryngectomy patients to recover their voices was described. The system comprised an articulator motion capture device [19], which monitored the movement of the lips and tongue by tracking the magnetic field generated by small magnets attached to them, and a synthesis module, which generated speech from articulatory data. To generate speech acoustics, recurrent neural networks (RNNs) trained on parallel articulatory and speech data were used. The speech produced by this system had a reasonable quality when evaluated on normal speakers, but it was not completely natural owing to limitations when estimating the pitch (i.e., the capturing device did not have access to any information about the glottal excitation).

In this work, we are interested in determining whether the proposed adapted SEGAN could improve those signals by generating more natural and realistic prosodic contours. To evaluate this, we have articulatory and speech data available, recorded simultaneously for 6 healthy British subjects (2 females and 4 males). Each speaker has recorded a random subset of the CMU Arctic corpus [20] (25 minutes for each speaker, approximately). Then, whispered speech was generated from the articulatory data by using the RNN-based articulatory-to-speech synthesiser described in [13]. In this work, these whispered signals are taken as the input to SEGAN, which acts as a post-filter enhancing the naturalness of the signals. For each whispered signal we have a natural version, which is the original speech signal recorded by the subject. To simplify our first modeling approach we used one male and one female speakers, namely M4 and F1, and built two speaker-dependent SEGAN models. These speakers are selected for the better level of intelligibility of their whisper data within their genders. We want to note, however, that both female speakers are less intelligible in their whisper form than male speakers. These two speakers’ data is split into two sets: (1) training, with approximately 90% of the utterances and (2) test, with the remaining approximate 10%. In order to have augmented data we follow the same chunking method as in our previous work [15] but window strides are one order of magnitude smaller. Hence we have a canvas of 16,384 samples (∼1 second at 16kHz) every 50 ms, in contrast with the previous 0.5 s.

#### 3.2. SEGAN Setup

We use the same kernel widths of 31 as we had in [15], both when encoding and decoding and for both $G$ and $D$ networks. The feature maps are incremental in the encoder and decremental in the decoder, having $\{64, 128, 256, 512, 1024, 512, 256, 128, 64, 1\}$ in the generator and $\{64, 128, 256, 512, 1024\}$ in the discriminator convolutional structures. The discriminator has a linear layer at the end with a single output neuron, as in the original SEGAN setup. The latent space is constructed with the concatenation of the thought vector $e \in \mathbb{R}_{T \times 1024}^{T \times 1024}$ with the noise vector $z \sim \mathcal{N}(0, I)$.

Both networks are trained with Adam [21] optimizer, with the two-timescale update rule (TTUR) [22], such that $D$ will have a four times faster learning rate to virtually emulate many iterations in $D$ prior to updating $G$. This way, we have $D$ learning rate 0.0004 and $G$ learning rate 0.0001, with $\beta_1 = 0$ and $\beta_2 = 0.9$, which are the same schedules based on recent successful approaches to faster and stable convergent adversarial training [23]. All signals processed by the GAN, either in the input/output of $G$ or the input of $D$, are pre-emphasized with a 0.95 factor, as it proved to help coping with some high-frequency artifacts in the de-noising setup. When we generate voiced data out of $G$ we de-emphasize it with the same factor to get the final result.
3.3. Baseline

To assess the performance of SEGAN in this task we have as reference the RNN-based articulatory-to-speech system from our previous work [13] and the natural data for each modeled speaker. The recurrent model is used to predict both the spectral (i.e., MFCCs) and pitch parameters (i.e., fundamental frequency, aperiodicities and unvoiced-voiced decision) from the articulatory data, so the source is articulatory data and not whispered speech in that case. The STRAIGHT vocoder [24] is then employed to synthesise the waveform from the predicted parameters.

4. Results

We analyze the statistics of the generated pitch contours for the RNN, SEGAN and natural data. Figure 3 depicts the histograms of all contours extracted from predicted/natural waveforms. Ahocoder [25] was used to extract logF0 curves, which are then converted to Hertz scale. Then, all voiced frames were selected and concatenated per each of the three systems. We come up with a long stream for each system and for the two genders. It can be seen that, for both genders, voiced histograms (corresponding to SEGAN) have a broader variance than RNN ones, closer to the natural signal shape. This is understandable if we consider that the RNN was trained with a regression criterion that optimizes its output towards the mean of the pitch distribution. This ends up producing a monotonic prosody effect, normally manifested as a robotic sounding that can be heard in the audio samples referenced below. This indicates that the adversarial procedure can generate more natural pitch values.

Figure 4 shows pitch contours generated by SEGAN with different random seeds. We have to note that each random seed generates a different latent vector z, so the stochasticity creates novel curves that look plausible. It also can be noted that SEGAN made some errors in determining the correct voicing decision for some speech segments. We may enforce a better behavior in a future version of the system with an auxiliary unvoiced/voiced classifier in the output of G.

Finally, figure 2 shows examples of waveforms and spectrograms for natural, whispered and voiced signals. We can appreciate how, for a small chunk of waveform, the generator network is able to refine low frequencies and gets rid of high frequency noises to approximate the natural data. Preliminary listening tests suggest that this model can achieve a good natural voiced version of the speech, but some artifacts intrinsic to the convolutional architecture (specially in high-frequencies) have to be palliated. This observation is in line with what was also prompted in the WaveGAN [26] work, and this is also one of the potential reasons of the effectiveness of using pre-emphasis. We refer the reader to the audio samples to have a feeling of the current quality of our system.

5. Conclusions

We presented a speaker-dependent end-to-end generative adversarial network to act as a post-filter of whispered speech to deal with a pathological application. We adapted our previous speech enhancement GAN architecture to overcome misalignment issues and still obtained a stable GAN architecture to reconstruct voiced speech. The model is able to generate novel pitch contours by only seeing the whispered version of the speech at its input. The method generates richer curves than the baseline, which sounds monotonic in terms of prosody. Future lines of work include an even more end-to-end approach by going sensor-to-speech. Also, further study is required to alleviate intrinsic high frequency artifacts provoked by the type of decimation-interpolation architecture we base our design on.

6. Acknowledgements

This research was supported by the project TEC2015-69266-P (MINECO/FEDER, UE).

http://veu.talp.cat/whispersegan/
7. References


LSTM based voice conversion for laryngectomees

Luis Serrano, David Tavarez, Xabier Sarasola, Sneha Raman, Ibon Saratxaga, Eva Navas, Inma Hernández

Aholab
University of the Basque Country (UPV/EHU), Bilbao, Spain
{lserrano, david, xsarasola, sneha, ibon, eva, inma}@aholab.ehu.eus

Abstract
This paper describes a voice conversion system designed with the aim of improving the intelligibility and pleasantness of oesophageal voices. Two different systems have been built, one to transform the spectral magnitude and another one for the fundamental frequency, both based on DNNs. Ahocoder has been used to extract the spectral information (mel cepstral coefficients) and a specific pitch extractor has been developed to calculate the fundamental frequency of the oesophageal voices. The cepstral coefficients are converted by means of an LSTM network. The conversion of the intonation curve is implemented through two different LSTM networks, one dedicated to the voiced unvoiced detection and another one for the prediction of F0 from the converted cepstral coefficients. The experiments described here involve conversion from one oesophageal speaker to a specific healthy voice. The intelligibility of the signals has been measured with a Kaldi based ASR system. A preference test has been implemented to evaluate the subjective preference of the obtained converted voices comparing them with the original oesophageal voice. The results show that spectral conversion improves ASR while restoring the intonation is preferred by human listeners.

Index Terms: voice conversion, speech and voice disorders, alaryngeal voices, speech intelligibility

1. Introduction
The laryngectomees are persons whose larynx has been surgically removed. The larynx is a fundamental organ in the production of speech since the vocal folds are located inside it. In spite of this, it is still possible to utter intelligible speech using alternative vibrating elements. The acquired new voice is called alaryngeal. There are three main ways to produce the alaryngeal voice: oesophageal (ES), electrolaryngeal (EL) and tracheoesophageal (TES) speech. The experiments described in this paper use only oesophageal speech.

Unlike the other two methods, the production of ES does not require any device. This kind of speech is learned with the help of a speech therapist. In this method the pharyngeoesophageal segment is used as a substitutive vibrating element for the vocal folds. Due to the nature of the intervention, the air used to create the vibration of the oesophagus can not come from the lungs and the trachea as happens during normal speech production. Instead, the air is swallowed from the mouth and introduced in the oesophagus, being then expelled in a controlled way while producing the vibration.

These huge differences in the production mechanisms lead to a diminution of naturalness and intelligibility [1, 2, 3]. As a consequence, the communication with others is hindered. Moreover, these less intelligible voices are an added problem for the automatic speech recognition algorithms that are becoming ubiquitous in the human computer interaction technologies. The work presented in this paper aims at improving the quality and intelligibility of ES, with the final purpose of contributing to a better life of the laryngectomy.

There have been different approaches to enhance the quality and the intelligibility of alaryngeal voices. Some research works use the source-filter analysis of the pathological signal and focus on the modifications of one or both source and filter. An example of this approach can be found in [4] where an adaptive gain equalizer algorithm is used to modify the source of ES; or in [5] where a reconstruction of normal sounding speech for laryngectomy patients is attempted through a modified CEP L codec. Another approach is to work with the prosodic elements. In [6], the pitch information extracted from an electroglottograph (EGG) is used to create a synthetic glottal signal, reducing jitter and shimmer. Additionally, spectral smoothing and tilt correction were applied. These modifications reduced the harshness and breathiness of the TE speech. The same authors relate in [7] a repairing method of the durations of the pathological phonemes. In the same line, [8] presents a system where concatenation of randomly chosen healthy reference patterns replaces the pathological excitation, adjusting the short, medium and long-term variability of the pitch.

A different approach to the problem is to make use of a voice conversion (VC) system. In a VC system the pursued goal is to transform utterances from a given source speaker into a specific target speaker, i.e., apply some techniques to perceive the sentences as uttered by the specific target speaker. For the purpose of enhancing alaryngeal voices, a healthy speaker is chosen as target speaker. Different examples of techniques based on statistical voice conversion can be found in [9], [10] or [11], where the characteristics of the target speaker can be tuned to obtain a more personalized converted voice.

The VC process can be divided into two stages: Firstly, a training phase is needed in order to learn the correspondences laying between source and target acoustic features. These learned relationships are then stored in the form of a conversion function. The second step is the conversion itself. The conversion function is applied to transform new input utterances from the source speaker. Although the identity of the speaker is also contained in the suprasegmental (prosody) and even linguistic features, VC research has been focused mostly on mapping spectral features [12, 13, 14].

The VC is a field that has been researched for a long time, an exhaustive recent review of the field can be found in [15]. A wide variety of approaches have been proposed to obtain the conversion function: from codebooks [16, 17] and hidden Markov models [18, 19, 20] to Gaussian Mixture Models (GMMs) [21, 12, 22, 23] or Gaussian processes [24]. In the last years, special focus has been given to shallow/deep neural networks (S/DNNs) solutions [25, 26, 14, 27, 28].
In this paper we propose to take advantage of the recent advances in machine learning techniques to train a deep learning system to convert from the audio of an oesophageal speaker to a healthy speaker. Though the literature focuses specially on spectral conversion, the importance of prosody in oesophageal speech can not be left out. A good intonation is important for the utterances to be perceived as more natural and pleasant. Therefore, the proposed system will not only convert the spectral features but will also estimate \( f_0 \). The spectral and \( f_0 \) conversion contributions have been then evaluated separately by means of word error rate (WER) and a perceptual test.

2. Voice conversion system

The proposed architecture uses two parallel DNN based systems to convert separately the spectral envelope and the fundamental frequency curve \( f_0 \). They will be described in detail in the following subsections.

The acoustic analysis is performed using Ahocoder [29]. This vocoder does the harmonic analysis of the speech audio files every 5 ms, extracting the Mel-cepstral (MCEP) representation of the spectral envelope, log\( f_0 \) and maximum voiced frequency (MVF). The MVF is related to the harmonicity of the signal and will not be converted (the MVF value obtained from the source signal is used). To extract \( f_0 \) from the alaryngeal signals, the default method of Ahocoder has been replaced by a more specific method, designed to cope with the irregularities present in these signals. This strategy mainly consists on using the autocorrelation method for pitch extraction over the residue signal of the PSIAIF analysis [30]. In this work 25 MCEP coefficients in combination with their first order derivatives are used for spectral conversion. In the experiment described here the 0\(^{th} \) coefficient (related to the energy), is not converted but directly copied from the source to the target. Similarly, the source MVF is not modified and it is directly copied to the target. Other alternatives such us using a constant MVF were also experimented but showed no significant differences in informal tests. For pitch prediction 40 MCEP coefficients are used. This number was chosen due to quality reasons observed during the training stages.

For the networks architecture, the chosen approach was to use a Long Short-Term Memory (LSTM) architecture. This type of neural network allows the net to retain (and gradually forget) information about previous time instants taking advantage of the strong temporal dependency that exists between consecutive frames in the speech signals, specially for the \( f_0 \) curve. The neural networks are implemented in Python using the Keras library.

2.1. Spectral conversion

Before training the LSTM network, it is necessary to align the source and target utterances. Due to the characteristics of the oesophageal speech, there is a very important mismatch between both healthy and oesophageal signals which causes the inadequacy of using a dynamic time warping (DTW) algorithm directly [31]. This is why both signals were labelled at phone level, and then the iterative alignment procedure described in [32] and [33] was applied for each pair of oesophageal and healthy phones.

With the alignment results, the inputs and outputs of the network are built. The corresponding first order derivatives are appended to the 24 source aligned cepstra \((c_0' \; c_0')\) (excluded). All vectors from all sentences are appended one after the other, resulting in two matrices for the source-target speaker pair. Then, mean and variance normalization is applied to each dimension of the cepstrum vector and its derivative. In addition, a voiced/unvoiced decision vector in the form of 0 or 1 obtained from the log\( f_0 \) values is appended, so the dimension of the input is finally 49. The resulting matrix is divided in sequences of 50 frames before entering the net.

The net will predict the 48 MCEP coefficients of the target speaker and the voiced/unvoiced decision vector. The metric we search to minimize is the MSE.

The number of cells of the LSTM layer is 100. As stated before, the number of frames that composes each batch sequence is 50, and the input dimension is equal to 49. The output is obtained from a fully connected layer and gives us a sequence of 50 frames and dimension 49 for each sequence in the input.

We used the RMSProp optimizer (divide the gradient by a running average of its recent magnitude) with a learning rate of 0.0001. To avoid overfitting, a dropout ratio of 0.5 was used. We trained the network for 60 epochs.

The performance of the network has been analysed comparing the predicted result with the target data. As the order of the coefficients goes up, the similarity between the predicted MCEP and the target one goes down. This can be seen by calculating the MSE for each normalized cepstral coefficient between the predicted and target MCEPs (Figure 1).

2.1.1. Maximum Likelihood Parameter Generation (MLPG)

In order to solve the problem of the smoothing present in the conversion process, we apply the maximum likelihood parameter generation algorithm [12]. We consider the MCEP obtained from the LSTM spectral conversion network as the mean vectors of a Gaussian distribution. Same as in [34], the covariance matrix is obtained from the squared error between the original features and the predicted converted vectors.

The global variance (GV) is calculated from the target training data and is used in the MLPG to do the spectral conversion along the mean vectors and covariance matrices for each test sentence.
2.2. Fundamental frequency estimation

Two different networks have been used to perform the estimation of the fundamental frequency, one for the prediction of \( f_0 \) and another one for U/V decision estimation (see Figure 2). In both cases, the LSTM model is trained to map the relationship between the healthy target MCEPs and their corresponding \( f_0 \) sequence. Mean and variance normalization is applied to each dimension of the cepstrum vector. \( \log f_0 \) is linearly interpolated in unvoiced frames and mean and variance normalization is also applied in this case. Development experiments showed that the use of 40 MCEP coefficients provided better results for the U/V estimation than using only 25 coefficients, therefore, 40 MCEP (no derivatives) are used for the intonation modelling.

For the U/V decision we use one Bidirectional LSTM layer with 64 cells. A fully connected layer with sigmoid activation provides the final output. The network is optimized using the Adam algorithm with minibatches of size 10 and trained for 60 epochs. A dropout ratio of 0.2 was applied as regularization. We employ the binary cross-entropy as loss function.

The same configuration is used for the \( f_0 \) prediction: one Bidirectional LSTM layer with 64 cells followed by a fully connected layer, with linear activation in this case. Also the MSE is now the metric we search to minimize, as it occurs for the cepstrum network.

In the conversion stage, 40 converted MCEP coefficients are needed. Therefore another network has been trained, similar to the one used for spectral conversion, to obtain the necessary number of coefficients. Finally, the \( \log f_0 \) is estimated from the converted MCEPs by the two conversion networks.

2.3. Training and testing data

Due to the nature of the data, the amount of data available for training and testing the conversion system is scarce. The parallel data used are the recordings of 100 phonetically balanced sentences selected from a bigger corpus [35] made by one healthy speaker and one oesophageal speaker. Out of the 100 utterances, 90 are chosen for training and 10 for testing the system, using 10-fold cross-validation.

3. Evaluation

3.1. Objective evaluation: Kaldi ASR

To have an objective measure (WER) of the effect of the conversion we have prepared an automatic speech recognizer (ASR) for Spanish using the Kaldi toolkit [36]. It is implemented following the recipe s5 for the Wall Street Journal database. The acoustic features used are 13 Mel-Frequency Cepstral Coefficients (MFCCs) to which a process of mean and variance normalization (CMVN) is applied to mitigate the effects of the channel.

The training begins with a flat-start initialization of context-independent phonetic Hidden Markov Models (HMM), and then a series of accumulative trainings are done. For the final step of the recognizer, a neural network is trained. The input features to the neural network consist of a series of 40-dimensional features. The network sees a window of these features, with 4 frames on each side of the central frame. The features are derived by processing the conventional 13-dimensional MFCCs. The necessary steps are described in [37], and basically consist in applying a series of transformations to the normalized cepstra: first linear discriminant analysis (LDA), then maximum likelihood linear transform (MMLT) and global feature-space maximum likelihood linear regression (IMLLR). At the recognition stage, the same transformations are applied to the test data, handling them as a block.

The audio material used to train the ASR is described in [38]. However, although the acoustic models have been maintained, the lexicon has been created from the 100 sentences corpus used in the experiment (701 words). This has been done because using the original lexicon (with 37,632 entries) as much as 23% of the words were out of vocabulary (OOV) words. This is due to the fact that the sentences are phonetically balanced and many sentences containing proper names and many unusual words were chosen to maximize the variability of the phonetic content. Together with this reduced lexicon, a unigram language model with equally probable words has been used. Although the final WER numbers obtained this way are not comparable to a realistic ASR situation, the procedure serves our purpose of evaluating the ability of the conversion system to improve (or not) the performing of any ASR system.

Three different recognition experiments have been carried out. The first one does the recognition of the 100 original sentences recorded by the laryngectomee speaker. The second one evaluates the resynthesized audio signals with Ahocoder using the original cepstrums and the estimated \( f_0 \) obtained from the network. The last one uses as input the 100 sentences resynthesized with the converted cepstrum and the estimated \( f_0 \).

The results are shown in Table 1. The WER of the original oesophageal signal is very high (56.93% vs 10.15% obtained for the target healthy speaker) showing the difficulty of the task. As expected, the system performs very similarly when only \( f_0 \) is changed. The small differences are probably due to the resynthesis process and recalculating the parameters from the waveforms performed by Kaldi. However, when the recognized signals are those which are resynthesized using the converted cepstrum and \( f_0 \), the WER drops by a 15% in absolute terms showing that some problems present in the oesophageal spectrum are corrected by the conversion system. The WER of the converted signals is though far away from the performance for the healthy voice.
Table 1: WER results for the different experiments

<table>
<thead>
<tr>
<th>Case</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>original healthy</td>
<td>10.15</td>
</tr>
<tr>
<td>original oesophageal</td>
<td>56.93</td>
</tr>
<tr>
<td>original spectrum + estimated ( f_0 )</td>
<td>57.91</td>
</tr>
<tr>
<td>converted spectrum + estimated ( f_0 )</td>
<td>41.48</td>
</tr>
</tbody>
</table>

3.2. Subjective evaluation: perceptual test

A perceptual listening test has been designed in order to evaluate the degree of preference over each processing technique (including no processing at all) by human evaluators. In this test, a total of 18 random sentences are selected for each listener, and for each of the sentences two of the three cases (original oesophageal (ORIGINAL), original spectrum + estimated \( f_0 \) and converted spectrum + estimated \( f_0 \)) are presented to the listener. He or she is asked to rate his or her preference over one of the two cases, by giving a score in a five point scale.

A total of 31 native Spanish speakers took part in the test. Figure 3 shows the averaged preference results of comparing the three systems in pairs. As it can be seen, the preferred system is the one with the original spectrum and modified intonation. The system that converts only \( f_0 \) is clearly preferred over the system with converted spectrum and \( f_0 \). Additionally, the original oesophageal signals are preferred over the signals with the converted spectrum. Thus, the signals with a converted spectrum are least preferred. This can be due to the loss of quality introduced by the conversion of the MCEP coefficients. Figure 4 shows the degree of preference for each pair of systems. The figure shows that the ‘strong preference’ option is the least chosen option in all three pairs. Also, it can be seen that the \( f_0 \)-only modification is the most preferred method, followed by no modification (original signals).

A few examples of the converted utterances can be found in the following website: http://aholab.ehu.eus/users/lserrano/ib18/demosib18.html.

4. Conclusions

In this paper we have presented a conversion system from one oesophageal speaker to one healthy speaker using a deep learning approach. We have trained an LSTM network to convert the spectral features of the speaker, and two BLSTMs to estimate the intonation curve: one to learn the underlying relationship existing between the healthy MCEPs and their corresponding \( \log f_0 \), and another to predict from the same healthy cepstrum the V/U decision. With the system implemented, we have evaluated the effects that the conversion has in comparison with the original pathological signal in terms of intelligibility and pleasantness. The effect of the conversion of the prosody has been evaluated separately from the complete conversion (spectral and \( f_0 \) conversion together).

The intelligibility has been evaluated by means of ASR, using the WER. The results show that using a healthy target speaker the recognition rate can be greatly improved (as much as an absolute gain of 15% in our experiment). The WER rate is still far away from that obtained for the healthy voices, but this demonstrates that the conversion of the spectral parameters is helping the oesophageal signal to be more similar to the healthy voices used to train the ASR system.

As expected, \( f_0 \)-only modification has little or no effect in the recognition rate. However, this modification was the most preferred by the listeners in the perceptual test, even over the original oesophageal voice with no modifications at all. This is an important result because it corroborates the relevance of having a restored source without modifying the speaker characteristics.

Once the deep learning conversion approach has been proved as valid, in the future we will better adjust the LSTM parameters, for example, the sequence size, to capture the prosody of a speaker in a more adequate way. We also plan to experiment different network architectures.

5. Acknowledgements

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness with FEDER support (RESTORE project, TEC2015-67163-C2-1-R), the Basque Government (BerbaOla project, KK-2018/00014) and from the European Unions H2020 research and innovation programme under the Marie Curie European Training Network ENRICH (675324).

6. References


Sign Language Gesture Classification Using Neural Networks

Zuzanna Parcheta¹, Carlos-D. Martínez-Hinarejos²

¹ Sciling S.L., Carrer del Riu 321, Pinedo, 46012, Spain
²Pattern Recognition and Human Language Technology Research Center, Universitat Politècnica de València, Camino de Vera, s/n, 46022, Spain

zparcheta@sciling.com, cmartine@dsic.upv.es

1. Introduction

Nowadays, technology has advanced to an unbelievable point in helping people with almost any kind of disability. People who are deaf or hard of hearing often experience limitations in their everyday life. However, assistive technology helps deaf people in their daily life. There are two main types of assistive devices for deafness: 1) Devices to enhance listening, e.g. frequency modulated (FM) systems [1], infrared (IR) systems [2], cochlear implants [3], and hearing aids [4] and 2) Devices to convey information visually, e.g. alerting devices [5], captioning [6], and real-time transcription systems [7]. Apart from that, most deaf people use sign language as their preferred way to communicate. However, there is no tool to help them interact with non-users of sign language. Fortunately, contemporary technology allows for developing systems of automatic translation from sign language into oral language.

To solve this problem it is necessary to create a tool which allows deaf people to communicate with non-users of sign language in a comfortable and fluid way. The researchers’ commitment to the deaf community is to improve the quality of the automatic systems that are used for this task by exploring new technologies and recognition patterns.

In this work, we explore Convolutional Neural Networks (CNN) to perform gesture classification for isolated words coming from a sign language database generated using the Leap Motion¹ sensor. Previous work [8, 9] demonstrated that it is possible to classify this dynamic gesture database using Hidden Markov Models with a 10.6% error rate. In our current work, this score is significantly improved to an error rate of 8.6%.

Abstract

Recent studies have demonstrated the power of neural networks for different fields of artificial intelligence. In most fields, such as machine translation or speech recognition, neural networks outperform previously used methods (Hidden Markov Models with Gaussian Mixtures, Statistical Machine Translation, etc.). In this paper, the efficiency of the LeNet convolutional neural network for isolated word sign language recognition is demonstrated. As a preprocessing step, we apply several techniques to obtain the same dimension for the input that contains gesture information. The performance of these preprocessing techniques on a Spanish Sign Language dataset is evaluated. These approaches outperform previously obtained results based on Hidden Markov Models.

Index Terms: gesture recognition, human-computer interaction, gesture classification, sign language, Leap Motion, Convolutional Neural Networks

2. Related work

The first step to developing data based gesture recognition systems is data acquisition. For data acquisition there are devices available which use accelerometers [10], electromyography (Myo’s armband physical principle) [11], or infrared light (Kinect’s and Leap Motion physical principle) [12, 13]. Also, to get gesture data it is possible to use cameras worn on an arm [14] or data gloves [15]. Publications about gestures recognition can be divided into two main groups depending on whether signs are static or dynamic.

There are quite a few studies about static sign recognition. In [16], video capture and Multilayer Perceptron (MLP) neural networks are used to recognise 23 static signs from the Colombian Sign Language. In [17], Gaussian Mixture Models are applied to recognise 5 static signs obtained from image captures.

Some authors also consider dynamic gestures recognition. In [18], the Microsoft Kinect is used to obtain features to identify 8 different signs using weighted Dynamic Time Warping (DTW). The latter approach is mentioned in [8], where a combination of $k$-Nearest Neighbour classifiers and DTW allows the recognition of 91 signs extracted by using the Leap Motion sensor. This work was continued in [9] where different typologies of Continuous Density Hidden Markov Models were applied. [19] describes a system to recognise very simple dynamic gestures which uses Normalised Dynamic Time Wrapping. In [20], authors introduce a new method to obtain gesture recognition, called Principal Motion Components (PMC). However, they do experiments with gestures not used in sign language.

Neural networks have gained a lot of interest in recent years. Consequently, deep architectures for gesture recognition have appeared. In [21], authors explore convolutional and bidirectional neural network architectures. In experiments, they use the Montalbano² dataset, which contains 20 Italian sign gesture categories. In [22], authors modified the Neural Machine Translation (NMT) system using the standard sequence to sequence framework to translate sign language from videos to text. Also in [23], the deep architecture is applied to tasks such as gesture recognition, activity recognition and continuous sign language recognition. The authors employ bidirectional recurrent neural networks, in the form of Long Short Term Memory (LSTM) units.

¹https://www.leapmotion.com/

²http://chalearnlap.cvc.uab.es/dataset/13/description/
In this work, Convolutional Networks are used to classify gestures from the Spanish sign language. Some improvements in recognition accuracy are obtained with respect to results from our previous studies [8, 9].

3. Experimental setup

3.1. Data preparation

The experimental data is the isolated gestures subset from the “Spanish sign language db”3, which is described in [8]. All gestures in this database are dynamic. Data was acquired using the Leap Motion sensor. Gestures are described with sequences of 21 variables per hand. This gives a sequence of 42 dimensional vector features per gesture. The isolated gesture subset is formed of samples corresponding to 92 words; for each word, 40 examples performed by 4 different people were acquired, giving a total number of 3,680 acquisitions. The dataset was divided into 4 balanced partitions with the purpose of conducting cross-validation experiments in the same fashion as those described in [9].

Data corresponding to each gesture is saved in a separated text file. Each row in the file contains information (42 feature values) about the gesture in a determined frame. Gestures have different numbers of frames; therefore, a preprocess step is required because we need fixed length data to train a neural network. Three different methods were used to fix the number of rows of each gesture to an equal length of matrix $max_{length}$:

- Trim data to fixed length (keeping the last $max_{length}$ vectors) or pad with 0 (at the beginning) to get the same number of rows.
- Our implementation of the trace segmentation technique [25]
- Linear interpolation of data using the `interp1d` method from the `scikit-learn` library.

3.2. Model

In our experiments we use the LeNet network [26]. LeNet is a convolutional network that has several applications, such as handwriting recognition or speech recognition. The architecture of this network is characterised to ensure invariance to some degree of shift, scale, and distortion.

In our case, the input plane receives the gesture data, and each unit in a layer receives inputs from a set of units located in a small neighbourhood in the previous layer. In Figure 1 the scheme of the LeNet architecture is shown. The first layer is a 2 dimensional convolution layer which learns convolution filters, where each filter is $20 \times 20$ units. After that, the ReLU activation function is applied and followed by a max-pooling with kernel size of 2. Once again, a convolution layer is applied, but this time with 20 convolution filters. The ReLU activation function is applied, previously applying max-pooling. A dropout layer with $p = 0.5$ is added and finally, a fully-connected layer is used. The number of input features depends on the $max_{length}$ parameter, and it is calculated by following Equation (1).

$$max_{length} = \frac{12}{4} \cdot 140$$  (1)

Table 1: Results with trim/zero padding preprocessing. Best results marked with “*”.

<table>
<thead>
<tr>
<th>Max_{Length}</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>97</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>12.0</td>
<td>13.1</td>
<td>12.0</td>
<td>12.5</td>
<td>11.6</td>
<td>12.7</td>
<td>12.6</td>
<td>11.4</td>
</tr>
<tr>
<td>10</td>
<td>11.0</td>
<td>11.0</td>
<td>10.1</td>
<td>11.6</td>
<td>10.6</td>
<td>11.8</td>
<td>11.4</td>
<td>11.2</td>
</tr>
<tr>
<td>15</td>
<td>10.8</td>
<td>11.2</td>
<td>9.8</td>
<td>11.3</td>
<td>9.8</td>
<td>11.4</td>
<td>11.1</td>
<td>10.3</td>
</tr>
<tr>
<td>20</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.9</td>
<td>9.4</td>
<td>11.1</td>
<td>10.8</td>
<td>9.9</td>
</tr>
<tr>
<td>25</td>
<td>10.4</td>
<td>10.0</td>
<td>9.6</td>
<td>10.9</td>
<td>9.5</td>
<td>10.5</td>
<td>11.1</td>
<td>10.2</td>
</tr>
<tr>
<td>30</td>
<td>11.1</td>
<td>10.6</td>
<td>9.4</td>
<td>10.6</td>
<td>9.3</td>
<td>11.9</td>
<td>11.2</td>
<td>10.3</td>
</tr>
<tr>
<td>35</td>
<td>10.9</td>
<td>10.7</td>
<td>10.0</td>
<td>10.5</td>
<td>9.4</td>
<td>11.0</td>
<td>10.3</td>
<td>10.2</td>
</tr>
<tr>
<td>40</td>
<td>10.7</td>
<td>10.4</td>
<td>9.9</td>
<td>10.5</td>
<td>9.3</td>
<td>10.8</td>
<td>10.3</td>
<td>9.9</td>
</tr>
<tr>
<td>45</td>
<td>11.2</td>
<td>10.1</td>
<td>10.4</td>
<td>10.5</td>
<td>9.7</td>
<td>11.0</td>
<td>10.3</td>
<td>10.1</td>
</tr>
<tr>
<td>50</td>
<td>11.2</td>
<td>10.1</td>
<td>10.3</td>
<td>10.5</td>
<td>9.6</td>
<td>10.5</td>
<td>10.3</td>
<td>10.2</td>
</tr>
</tbody>
</table>

The number of output features is fixed to 1,000. Finally, there is an output layer with 92 neurons corresponding to the number of classes.

4. Experiments and results

We conducted experiments using our LeNet network. In each experiment we varied the value of patience $p = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$ and $max_{length} = \{40, 50, 60, 70, 80, 90, 97, 100\}$. Patience (also called early stopping) is a technique for controlling overfitting in neural networks, by stopping training before the weights have converged. We stop the training when the performance has stopped improving in a determined number of epochs. The original code and LeNet configuration is dedicated to speech recognition, where the maximum length was fixed to 97 by default. That is the reason why this value is used in the experiments. The experiments compare against the baseline provided by [9], which used Hidden Markov Models (HMM) to obtain a classification error rate of $10.6 \pm 0.9$. Confidence intervals were calculated by using the bootstrapping method with 10,000 repetitions [27] and they are all around the same value for all the experiments ($\pm 0.9$). All the results shown in the following are classification error rates obtained through cross-validation using the same four partitions stated in [9].

For each experiment we used the three different techniques of data preprocessing described above. Table 1 shows gesture classification results using the trim/zero padding preprocessing technique. We added a colour scale to make it easier to see the performance of classification depending on patience and $max_{length}$ values. Dark colours are for bigger error rates and light colours are assigned to lower error rates. We can appreciate that the best score is obtained with a patience of 30 or 40 epochs and $max_{length}$ of 80 rows (9.3% error rate). In general, the higher the patience, the lower the error, but only until a certain value. With respect to $max_{length}$, the behaviour does not present a clear pattern.

In Table 2 we show most frequent confused gestures in gesture classification using trim/zero padding as preprocessing step. The confusions comes from global confusion matrix generated during cross-validation experiments.

In Table 3, trace-segmentation preprocessing results are shown. In this case, the best score is obtained with a patience of 45 epochs and $max_{length}$ of 100 rows (9.2% of error rate). This result is slightly better than the best score obtained with the trim/zero padding technique. However, according to confidence

https://github.com/Sasanita/spanish-sign-language-db
Table 2: Most frequent confused gestures in gesture classification using trim/zero padding preprocessing.

<table>
<thead>
<tr>
<th>Num of confusions</th>
<th>Reference</th>
<th>Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>sign</td>
<td>a lot</td>
</tr>
<tr>
<td>9</td>
<td>thank you</td>
<td>nineteen</td>
</tr>
<tr>
<td>8</td>
<td>thirteen</td>
<td>fourteen</td>
</tr>
<tr>
<td>8</td>
<td>sign</td>
<td>regular</td>
</tr>
<tr>
<td>7</td>
<td>you (male)</td>
<td>we (male)</td>
</tr>
<tr>
<td>7</td>
<td>one</td>
<td>eleven</td>
</tr>
<tr>
<td>7</td>
<td>red</td>
<td>eyes</td>
</tr>
</tbody>
</table>

Table 3: Results with trace-segmentation preprocessing. Best result marked with “*”.

<table>
<thead>
<tr>
<th>Max ( \text{length} )</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>97</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patience</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>12.5</td>
<td>12.6</td>
<td>13.2</td>
<td>11.9</td>
<td>11.5</td>
<td>11.1</td>
<td>12.4</td>
<td>11.4</td>
</tr>
<tr>
<td>10</td>
<td>11.4</td>
<td>10.8</td>
<td>11.5</td>
<td>10.5</td>
<td>12.6</td>
<td>11.7</td>
<td>10.7</td>
<td>11.1</td>
</tr>
<tr>
<td>15</td>
<td>11.2</td>
<td>11.3</td>
<td>11.3</td>
<td>10.6</td>
<td>10.3</td>
<td>10.9</td>
<td>10.0</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>11.1</td>
<td>9.8</td>
<td>10.7</td>
<td>10.5</td>
<td>10.8</td>
<td>10.2</td>
<td>10.0</td>
<td>11.0</td>
</tr>
<tr>
<td>25</td>
<td>10.2</td>
<td>9.7</td>
<td>10.3</td>
<td>10.1</td>
<td>10.7</td>
<td>10.1</td>
<td>9.7</td>
<td>9.7</td>
</tr>
<tr>
<td>30</td>
<td>10.7</td>
<td>9.7</td>
<td>9.9</td>
<td>10.4</td>
<td>11.1</td>
<td>10.2</td>
<td>9.7</td>
<td>9.8</td>
</tr>
<tr>
<td>35</td>
<td>10.7</td>
<td>9.3</td>
<td>9.9</td>
<td>10.5</td>
<td>10.7</td>
<td>10.3</td>
<td>9.3</td>
<td>9.6</td>
</tr>
<tr>
<td>40</td>
<td>10.1</td>
<td>9.7</td>
<td>10.0</td>
<td>10.1</td>
<td>10.0</td>
<td>9.5</td>
<td>9.6</td>
<td>9.7</td>
</tr>
<tr>
<td>45</td>
<td>10.0</td>
<td>9.9</td>
<td>10.0</td>
<td>10.7</td>
<td>10.1</td>
<td>10.0</td>
<td>9.4</td>
<td><em>9.2</em></td>
</tr>
<tr>
<td>50</td>
<td>10.2</td>
<td>9.6</td>
<td>10.0</td>
<td>10.1</td>
<td>10.1</td>
<td>10.2</td>
<td>9.6</td>
<td>9.8</td>
</tr>
</tbody>
</table>

In Table 4 we show most frequent confused gestures in gesture classification using trace-segmentation preprocessing step. Some of confused gestures match with confusions from Table 2.

<table>
<thead>
<tr>
<th>Num of confusions</th>
<th>Reference</th>
<th>Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>one</td>
<td>no</td>
</tr>
<tr>
<td>10</td>
<td>sign</td>
<td>a lot</td>
</tr>
<tr>
<td>10</td>
<td>red</td>
<td>eyes</td>
</tr>
<tr>
<td>10</td>
<td>gray</td>
<td>colour</td>
</tr>
<tr>
<td>9</td>
<td>no</td>
<td>one</td>
</tr>
<tr>
<td>9</td>
<td>thank you</td>
<td>study</td>
</tr>
<tr>
<td>8</td>
<td>be born</td>
<td>good morning</td>
</tr>
<tr>
<td>8</td>
<td>eighteen</td>
<td>nineteen</td>
</tr>
<tr>
<td>7</td>
<td>eyes</td>
<td>red</td>
</tr>
<tr>
<td>7</td>
<td>five</td>
<td>hello</td>
</tr>
</tbody>
</table>

The last experiments were conducted using the interpolation technique as a preprocessing step. This time the best result was obtained with a patience of 25 epochs and 97 for \( \text{max} \text{.length} \). These are definitely the best results, since the lowest error rate is 8.6%, which is significantly better than the HMM baseline (which did not happen with the other preprocessing techniques). Therefore, the previous result from [9] is improved by about 2% in an absolute manner. This result is considered to be very satisfactory.

In Table 6 we show most frequent confused gestures in gesture classification using interpolation as preprocessing step.

We can appreciate that some of confusions match in each of confusion table e.g. “sign” confused with “a lot” sign. In most of confused signs there are some similarity in shape of hands or trajectory during gesture e.g. sign “hello” and “five” have the same shape of hand only that the sign “hello” make swinging movement. We show image of both signs in Figure 2. Exactly the same thing happens with signs “one” and “no” where the only difference between them is swinging movement.

5. Conclusion and future work

In this work, we developed a classification system for the Spanish sign language. We used the LeNet convolutional network. As a preprocessing step we employed three different methods. The data set used in this work is the “Spanish sign language db” which contains data from 92 different gestures captured with the Leap Motion sensor. The results obtained are very satisfactory, since we were able to improve our baseline by lowering

Table 5: Results with interpolation preprocessing. Best result marked with “*”.

<table>
<thead>
<tr>
<th>Max_length</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>97</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patience</td>
<td>130</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>12.1</td>
<td>11.4</td>
<td>12.3</td>
<td>11.6</td>
<td>12.0</td>
<td>11.3</td>
<td>10.7</td>
<td>12.7</td>
</tr>
<tr>
<td>10</td>
<td>11.1</td>
<td>11.4</td>
<td>11.2</td>
<td>10.6</td>
<td>10.9</td>
<td>10.1</td>
<td>9.9</td>
<td>10.6</td>
</tr>
<tr>
<td>15</td>
<td>10.3</td>
<td>9.5</td>
<td>11.9</td>
<td>10.0</td>
<td>10.3</td>
<td>10.1</td>
<td>10.1</td>
<td>9.2</td>
</tr>
<tr>
<td>20</td>
<td>10.5</td>
<td>9.3</td>
<td>10.3</td>
<td>9.4</td>
<td>11.4</td>
<td>10.0</td>
<td>9.5</td>
<td>9.0</td>
</tr>
<tr>
<td>25</td>
<td>9.6</td>
<td>9.8</td>
<td>9.9</td>
<td>9.5</td>
<td>9.8</td>
<td>9.9</td>
<td><em>8.6</em></td>
<td>9.6</td>
</tr>
<tr>
<td>30</td>
<td>10.0</td>
<td>10.1</td>
<td>10.3</td>
<td>9.9</td>
<td>9.6</td>
<td>9.7</td>
<td>8.9</td>
<td>9.1</td>
</tr>
<tr>
<td>35</td>
<td>9.7</td>
<td>10.0</td>
<td>9.8</td>
<td>9.6</td>
<td>9.8</td>
<td>9.2</td>
<td>8.7</td>
<td>9.0</td>
</tr>
<tr>
<td>40</td>
<td>10.0</td>
<td>9.7</td>
<td>10.0</td>
<td>9.6</td>
<td>9.6</td>
<td>9.2</td>
<td>8.9</td>
<td>9.0</td>
</tr>
<tr>
<td>45</td>
<td>10.0</td>
<td>9.7</td>
<td>9.9</td>
<td>9.4</td>
<td>9.8</td>
<td>9.5</td>
<td>9.2</td>
<td>9.6</td>
</tr>
<tr>
<td>50</td>
<td>10.0</td>
<td>9.7</td>
<td>9.9</td>
<td>9.5</td>
<td>9.6</td>
<td>9.5</td>
<td>9.1</td>
<td>9.0</td>
</tr>
</tbody>
</table>

Table 6: Most frequent confused gestures in gesture classification using interpolation preprocessing.

<table>
<thead>
<tr>
<th>Num of confusions</th>
<th>Reference</th>
<th>Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>sign</td>
<td>a lot</td>
</tr>
<tr>
<td>10</td>
<td>eighteen</td>
<td>nineteen</td>
</tr>
<tr>
<td>9</td>
<td>red</td>
<td>eyes</td>
</tr>
<tr>
<td>9</td>
<td>he</td>
<td>brother</td>
</tr>
<tr>
<td>8</td>
<td>good morning</td>
<td>be born</td>
</tr>
<tr>
<td>7</td>
<td>one</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>do not know</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>hello</td>
<td>five</td>
</tr>
</tbody>
</table>

the error rate by 2% to a level of 8.6%. We also present tables containing most frequent confused gestures using different methods of preprocessing.

As we obtained such a good result, we would like to continue working in this area. Now we want to work with sign language sentence recognition to be able to create a system that recognises a signed sentence composed of several signs. For that, we will use Recurrent Neural Networks (RNN) commonly used in machine translation and speech recognition. Also, we will conduct experiments on isolated gestures classification to compare the performance of LeNet and RNN.

6. Acknowledgements

Work partially supported by MINECO under grant DI-15-08169, by Sciling under its R+D programme, and by MINECO/FEDER under project CoMUN-HaT (TIN2015-70924-C2-1-R). The authors would like to thank NVIDIA for their donation of Titan Xp GPU that allowed to conduct this research.

7. References


Influence of tense, modal and lax phonation on the three-dimensional finite element synthesis of vowel [a]

Marc Freixes, Marc Arnela, Joan Claudi Socoró, Francesc Alías, Oriol Guasch
GTM – Grup de Recerca en Tecnologies Mèdia, La Salle - Universitat Ramon Llull
Quatre Camins, 30, 08022 Barcelona, Spain
{marc.freixes,marc.arnela,joanclaudi.socoro,francesc.alias,oriol.guasch}@salle.url.edu

Abstract

One-dimensional articulatory speech models have long been used to generate synthetic voice. These models assume plane wave propagation within the vocal tract, which holds for frequencies up to ~5 kHz. However, higher order modes also propagate beyond this limit, which may be relevant to produce a more natural voice. Such modes could be especially important for phonation types with significant high frequency energy (HFE) content. In this work, we study the influence of tense, modal and lax phonation on the synthesis of vowel [a] through 3D finite element modelling (FEM). The three phonation types are reproduced with an LF (Liljencrants-Fant) model controlled by the $R_d$ glottal shape parameter. The onset of the higher order modes essentially depends on the vocal tract geometry. Two of them are considered, a realistic vocal tract obtained from MRI and a simplified straight duct with varying circular cross-sections. Long-term average spectra are computed from the FEM synthesised [a] vowels, extracting the overall sound pressure level and the HFE level in the 8 kHz octave band. Results indicate that higher order modes may be perceptually relevant for the tense and modal voice qualities, but not for the lax phonation.

Index Terms: voice production, higher order modes, high frequency energy, glottal source modelling, LF model, numerical simulation, finite element method

1. Introduction

For many years, works on articulatory speech synthesis have considered a simplified one-dimensional (1D) representation of the vocal tract. This is built from the so-called vocal tract area functions, which describe the cross-sectional area variations along the vocal tract center midline (see e.g., [1]). Voice is then synthesised by simulating the propagation of acoustic waves within this 1D representation of the vocal tract (see e.g., [2, 3, 4]). However, 1D approaches assume plane wave propagation, so they can only correctly approximate the acoustic waves within this 1D representation of the vocal tract (see e.g., [1]).

Beyond this limit, not only planar modes get excited but also higher order propagation modes appear, which strongly change the high frequency energy (HFE) content of the spectrum [5, 6] compared to that from a 1D model. Although the high frequency range has not received much attention in the literature, some recent studies point out that the HFE may be relevant for voice quality, speech localisation, speaker recognition and intelligibility (see [7] and references therein).

On the other hand, three-dimensional (3D) models do not need to assume plane wave propagation, since they can directly deal with 3D vocal tract geometries to emulate the complex acoustic field generated during voice production [8, 9, 10]. However, higher order modes do not always appear even if a 3D acoustic model is used. As shown in [6], a straightened vocal tract based on circular cross-sections prevents the onset of such modes due to radial symmetry, in contrast to what occurs for realistic vocal tract geometries based on MRI data. Other vocal tract geometries simplifications were studied in that work, all of them showing large variations in the HFE while keeping a similar behaviour for low frequencies. One can then assert that the vocal tract shape is determinant for the HFE content of the generated sound. However, the vocal tract shape is not the only factor affecting the HFE. The type of phonation can also modify it, as shown for instance in [11] for sustained vowels with loud and soft phonation.

In this work we study the effect of tense, modal and lax phonation on the synthesis of vowel [a], paying special attention to the HFE content. These three phonation types are reproduced using an LF (Liljencrants-Fant) model [12]. Although this model cannot consider the interaction between the vocal tract and the vocal folds [13, 14], it has proved to be useful to explore the phonatory tense-lax continuum [15] by controlling the $R_d$ glottal shape parameter [16]. Regarding the vocal tract, we consider an MRI-based realistic geometry, and its simplified counterpart considering circular cross-sections in a straightened midline [6]. This allows us to somewhat “activate” and “deactivate” the higher order modes. Different versions of vowel [a] are generated by convolving the LF glottal source signals with the vocal tract impulses responses obtained using a 3D acoustic model based on the Finite Element Method (FEM) [17]. In order to analyse the relevance of the higher order propagation modes for the lax, modal and tense phonation, the long-term average spectra (LTAS) and the HFE levels of the synthesised vowels are computed and compared.

The paper is structured as follows. The methodology used to study the production of vowel [a] with the three phonation types and the two vocal tract geometries is explained in Section 2. Next, the obtained results are discussed in Section 3. Finally, conclusions and future work are presented in Section 4.

2. Methodology

Figure 1 depicts the process followed to synthesise six versions of vowel [a]. These were obtained by convolving three glottal source signals with the FEM impulse responses of two vocal tract geometries that produce this vowel. In particular, and as mentioned before, we used the realistic vocal tract and the simplified straightened simplification with circular cross-sections from [6] (see Section 2.1), and computed their impulse responses $h(t)$ using the FEM (see Section 2.2). The glottal source signals $u_{gl}(t)$ were generated by means of an $R_d$ controlled LF model. The values $R_d = 0.3, 1, 2.7$ were selected from the $R_d$ range [0.3, 2.7] (see [16]) to reproduce a tense, a modal, and a lax phonation, respectively (see Section 2.3).

132

10.21437/IberSPEECH.2018-28
Figure 1: Synthesis of vowel [A] with a realistic vocal tract geometry (above) and its simplified counterpart of circular cross-sections in a straightened midline (below). Three phonation types are considered to reproduce a tense (dashed red line), a modal (solid black line) and a lax (dotted green line) voice production. The output pressure signal $p(t)$ is computed as the convolution of the glottal source $u_g(t)$ with the vocal tract impulse response $h(t)$ obtained from 3D FEM simulations.

For each vowel, the LTAS was computed as the Welch’s power spectral density estimate, with a 15 ms hammer window, 50% overlap and a 2048-point FFT. The overall energy levels and the HFE levels in the 8 kHz octave band were also extracted as in [11]. The 16 kHz octave band was not considered, since HFE changes in this frequency range were found almost perceptually irrelevant in [11].

### 2.1. Vocal Tract Geometries

Two vocal tract geometry simplifications of vowel [a] have been employed in this work, namely, the realistic configuration and the simplified straight vocal tract with circular shape (see Fig. 1). These geometries were obtained in [6] by simplifying the MRI-based vocal tract geometries in [18]. In a nutshell, the procedure consisted in the following. First, the subglottal tube, the face and the lips were removed from the original geometry (see [10] for a detailed analysis of the lips influence on simulations). Moreover, side branches such as the piriform fossae and valleculae were occluded (see e.g. [9, 19] for their acoustic effects). Cross-sections were next extracted as typically done to generate 1D area functions, but preserving their shapes and locations in the vocal tract midline. The realistic configuration was generated by linearly interpolating the resulting cross-sections. As shown in [6], this simplification provides very similar results to the original MRI-based vocal tract geometry without branches.

In the simplified straight vocal tract configuration, the cross-sectional shapes were modified to be that of a circle, preserving the same area. These circular cross-sections were located in a straightened version of the vocal tract midline and then linearly interpolated. The two configurations are hereafter referred as the realistic and the simplified vocal tracts.

### 2.2. Vocal Tract impulse response

The impulse response of each vocal tract geometry was computed using a custom finite element code that numerically solves the acoustic wave equation,

$$\partial^2_{tt}p - c^2_0\nabla^2p = 0,$$

combined with a Perfectly Matched Layer (PML) to account for free-field propagation [17]. In Eq. (1) $p(x, t)$ is the acoustic pressure, $\partial^2_{tt}$ stands for the second order time derivative, and $c_0$ is the speed of sound which is set to the usual value of 343 m/s. A Gaussian pulse was introduced on the glottal cross-sectional area as an input volume velocity $u_g(t)$. This pulse is of the type

$$u_g(t) = e^{-\left[(t-T_{gp})/0.29T_{gp}\right]^2} \text{[m}^3/\text{s}],$$

with $T_{gp} = 0.646/f_c$ and $f_c = 10$ kHz. Wall losses were considered by imposing a boundary admittance coefficient of $\mu = 0.005$ on the vocal tract walls. A 20 ms simulation was then performed capturing the acoustic pressure $p(t)$ at a node located outside of the vocal tract, 4 cm away from the mouth aperture center. The sampling frequency was set to $f_s = 8000$ kHz, which ensures a restrictive stability condition of the Courant-Friedrichs-Levy type required by explicit numerical schemes (see [17] for details on the numerical scheme).

A vocal tract transfer function $H(f)$ was computed from each simulation to compensate for the slight energy decay in frequency of the Gaussian pulse. This is defined as

$$H(f) = \frac{P_r(f)}{U_g(f)},$$

with $P_r(f)$ and $U_g(f)$ being the Fourier Transform of $p_r(t)$ and $u_g(t)$, respectively. $H(f)$ was computed up to 12 kHz, to allow the calculation of HFE level in the 8 kHz octave band [11]. The vocal tract transfer functions $H(f)$ for the realistic and the simplified geometries of vowel [a] are shown in Fig. 2 (also reported in [6], but only up to 10 kHz). As can be observed, planar modes are mainly produced below 5 kHz giving place to the first vowel formants. Beyond this value, higher order modes can also propagate, resulting in the more complex spectrum of vowel formants. Beyond this value, higher order modes can also propagate, resulting in the more complex spectrum of vowel formants. Beyond this value, higher order modes can also propagate, resulting in the more complex spectrum of vowel formants. Beyond this value, higher order modes can also propagate, resulting in the more complex spectrum of vowel formants.

Figure 2: Vocal tract transfer function $H(f)$ for the vowel [a] with the realistic and simplified vocal tract geometries.
the realistic geometry. Note, however, that these modes do not appear in the spectrum of the simplified configuration. The radial symmetry of this geometry prevents their onset [5, 6].

Finally, the inverse Discrete Fourier Transform was applied to the vocal tract transfer functions \( H(f) \) to obtain the vocal tract impulse responses \( h(t) \) of the two geometries (see Fig. 1).

### 2.3. Voice Source Signal

An LF model [12] was used to produce the voice source signal. This model approximates the glottal flow \( u_g(t) \) and its time derivative \( u'_g(t) \) in terms of four parameters \( (T_p, T_e, T_r, E_r) \) that describe its time-domain properties (see Fig. 3). The control of this model can be simplified with the single glottal shape parameter \( R_d \) [16]. This is defined as

\[
R_d = \frac{T_d}{T_s} \cdot \frac{1}{1 + \frac{U_0}{E_x} \cdot \frac{F_0}{F_s}},
\]

where \( T_d \) is the declination time, \( T_s \) the period, and \( F_s \) the fundamental frequency. The declination time \( T_d \) corresponds to the quotient between the glottal flow peak \( U_0 \) and the negative amplitude of the differentiated glottal flow \( E_x \).

In this work, we used the Kawahara’s implementation of the LF model [20], which generates a free-aliasing excitation source signal. We adapted this model to our purposes, modifying the sampling frequency from its original value of 44100 Hz to 24 kHz. Moreover, we introduced the \( R_d \) glottal shape parameter. This allows one to easily control the voice source with a single parameter, which runs from \( R_d = 0.3 \) for a very abducted phonation, to \( R_d = 2.7 \) for a very abducted phonation (see [16]). From the \( R_d \) range [0.3, 2.7] two extreme values plus a middle one were chosen. We used \( R_d = 0.3 \) to generate a tense phonation, \( R_d = 2.7 \) for a lax production, and \( R_d = 1 \) for a normal (modal) voice quality. With regard to \( F_0 \), a pitch curve was obtained from a real sustained vowel lasting 4.4 seconds. This pitch contour was placed around 120 Hz to generate all the source signals. Figure 4a shows four periods of the three simulated voice source waveforms. Moreover, the LTAS of the glottal source signals are represented in Fig. 4b. As observed, the phonation type obviously changes the glottal pulse shape, thus modifying the spectral energy distribution of the source signal.

### 3. Results

Six versions of vowel \( [a] \) (see Fig. 1) have been generated using the three glottal source signals corresponding to a tense, a modal and a lax phonation, and the two impulse responses obtained from the 3D FEM simulations of the realistic and simplified vocal tract geometries. The six synthesised vowels are normalised with the same scaling factor to obtain reasonable sound pressure levels. This factor has been selected so as to produce 70 dB SPL in the realistic geometry with a modal phonation \( (R_d = 1) \). The LTAS have then been computed for each audio.

Figure 5 shows the obtained LTAS for the six generated vowels. As also appreciated in the vocal tract transfer functions (see Fig. 2), small differences between geometries are produced for frequencies below 5 kHz, whereas beyond this range higher order modes propagate in the realistic case, thus inducing larger deviations. This behaviour can be observed for the three phonation types. Essentially the glottal source modifies the overall energy level and also introduces an energy decay in frequency (compare Fig. 2 with Fig. 5). This decay, known as the spectral tilt, strongly depends on the phonation type. The laxer the phonation the larger the spectral tilt [16]. Furthermore, the voice source also affects the energy balance of the first harmonic (below ~ 500 Hz). For instance, the lax phonation has the lowest overall energy values among all phonation types. However, one can see that the first harmonic (close to 120 Hz) has larger amplitude levels than the rest of the spectrum, in contrast to what occurs for the other phonations.

HFE levels have been computed by integrating the power spectral density in the 8 kHz octave band, as in [11]. In addition, the overall energy levels have been calculated following the same procedure but for the whole examined frequency range. The obtained results are listed in Table 1. Note first that in the realistic case with a modal phonation \( (R_d = 1) \) the overall level is 70 dB SPL. Remember that this value was fixed to
compute the scaling factor used to normalise the audio files. The overall level variations for the other configurations will thus correspond to modifications either introduced by the vocal tract geometry or by the glottal source. As expected, the larger the $R_d$ value (laxer phonation) the smaller the overall levels.

Far more interesting is to compare the results between geometries. The HFE levels decay between 5.6 dB and 5.9 dB for the realistic vocal tract depending on the phonation type, which only manifests as an overall level difference of 1.2 dB and 1.4 dB. The higher order modes tend to reduce the levels in the HFE content. According to [11], minimum difference limen scores of about 1 dB are given for normal-hearing listeners in the 8 kHz octave band, so one may hypothesise that the higher order modes may be perceptually relevant. However, depending on the phonation type the HFE could be too small to notice any difference. This seems to be the case of the lax phonation ($R_d = 2.7$), which gives HFE levels of 1.0 dB and 6.6 dB, depending on the geometry. We may then conjecture, that for this phonation type no differences in the outputs from the two geometries will be perceived. In other words, we would not notice the influence of higher order modes.

### Table 1: Overall and High-Frequency Energy (HFE) levels (in dB) obtained in the realistic and simplified vocal tract configurations of vowel [a] with a tense ($R_d = 0.3$), a modal ($R_d = 1$), and a lax ($R_d = 2.7$) phonation. $\Delta$ denotes the difference between the two vocal tract geometries.

<table>
<thead>
<tr>
<th>$R_d$</th>
<th>Geometry</th>
<th>Overall</th>
<th>$\Delta$ Overall</th>
<th>HFE</th>
<th>$\Delta$HFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>realistic</td>
<td>82.2</td>
<td>1.2</td>
<td>41.4</td>
<td>5.8</td>
</tr>
<tr>
<td></td>
<td>simplified</td>
<td>83.4</td>
<td></td>
<td>47.3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>realistic</td>
<td>70.0</td>
<td>1.3</td>
<td>14.9</td>
<td>5.9</td>
</tr>
<tr>
<td></td>
<td>simplified</td>
<td>71.3</td>
<td></td>
<td>20.8</td>
<td></td>
</tr>
<tr>
<td>2.7</td>
<td>realistic</td>
<td>63.5</td>
<td>1.4</td>
<td>1.0</td>
<td>5.6</td>
</tr>
<tr>
<td></td>
<td>simplified</td>
<td>64.9</td>
<td></td>
<td>6.6</td>
<td></td>
</tr>
</tbody>
</table>

4. Conclusions

In this work we have studied the influence of tense, modal and lax phonation on the 3D finite element synthesis of vowel [a], considering a realistic and a simplified vocal tract geometry. The 3D simulations behave very similarly for both geometries below 5 kHz, but significant differences appear beyond this frequency because of the rising of higher order propagation modes. It is worth mentioning that these modes only appear when using the realistic vocal tract. They induce a reduction of the HFE levels at the 8 kHz octave band from 5.6 to 5.9 dB, depending on the phonation type. These differences may be perceptually relevant, according to previous works in the literature. Specifically, a realistic 3D vocal tract geometry would be required for an accurate synthesis of vowel [a] through 3D FEM, when trying to simulate a modal and a tense voice production. Conversely, when a lax phonation is considered, the influence of higher order propagation may be imperceptible, since the HFE levels are very small. Therefore, a simpler 1D simulation would suffice in this case.

Future work will consider other $R_d$ values and geometry simplifications as well as other vowels to complete the study. Finally, we also plan to include aspiration noise in the LF model to evaluate its impact on the HFE content of the numerical simulations.

5. Acknowledgements

The authors are grateful to Saeed Dabbaghchian for the design of the vocal tract geometry simplifications. This research has been supported by the Agencia Estatal de Investigación (AEI) and FEDER, EU, through project GENIOVOX TEC2016-81107-P. The fourth author acknowledges the support from the Obra Social “La Caixa” under grant ref. 2018-URL-IR1rQ-021.

6. References


Exploring Advances in Real-time MRI for studies of European Portuguese

Conceição Cunha¹, Samuel Silva², António Teixeira², Catarina Oliveira³, Paula Martins⁴, Arun Joseph⁵, Jens Frahm⁵

¹IPS, Ludwig-Maximilians-Universität, Germany
²DETI/IEETA, Universidade de Aveiro, Portugal
³ESSUA/IEETA, Universidade de Aveiro, Portugal
⁴ESSUA/IBIMED/IEETA, Universidade de Aveiro, Portugal
⁵Max Plank Institute for Biophysical Chemistry, Germany

cunha@phonetik.uni-muenchen.de, sss@ua.pt, ajst@ua.pt, coliveira@ua.pt, pmartins@ua.pt, arun-antony.joseph@mpibpc.mpg.de, jfrahm@gwdg.de

Abstract

Recent advances in real-time magnetic resonance imaging (RT-MRI) for speech studies, providing a considerable increase in time resolution, potentially improve our ability to study the dynamic aspects of speech production. To take advantage of the sheer amount of the resulting data, automated methods can be used to select, process, and analyze the data, and previous work could tackle these challenges for an European Portuguese (EP) corpus acquired at 14 frames per second (fps).

Aiming to further explore RT-MRI in the study of the dynamic characteristics of EP sounds, e.g., nasal vowels and diphthongs, we present a novel 50 fps RT-MRI corpus and assess the applicability, in this new context, of our previous proposals for processing and analyzing these data to extract relevant articulatory information. Importantly, at this stage, we were interested in assessing if and to what extent the new data and the proposed methods are able to support and corroborate the articulatory analysis obtained from the previous corpus. Overall, and although this new corpus poses novel challenges, it was possible to process and analyze the 50 fps data. A comparison of automated analysis performed for the same sounds, for both corpora (i.e., 14 fps and 50 fps), yields similar results, corroborating previous results and demonstrating the envisaged replicability. Moreover, we updated the processing in order to be able to analyze dynamic information and provide first insights on the temporal organization of complex sounds such as nasal vowels and diphthongs.

Index Terms: speech production, dynamic, real-time magnetic resonance, European Portuguese, processing and analysis.

1. Introduction

A major challenge in modern linguistics is to understand how continuous and dynamic speech movements are related to perceptual categories like consonants and vowels. Physiological studies have shown that phonological segments can be defined based on the synchronization of primary articulators in speech (the lips, tongue, soft-palate, jaw, and vocal folds) with each other in time [1] using methods such as Electromagnetic Articulography (EMA) and Magnetic Resonance Imaging (MRI).

Many advances in real-time magnetic resonance imaging (RT-MRI) resolutions (spatial and temporal) have been driven by the need to investigate phonetic and phonological phenomena [2], such as vowel nasalization in Portuguese [3]. In the beginning of RT-MRI application to the study of speech production the temporal resolution was quite low, but this application was already successful in providing for the first time information about the entire tongue contour instead of some points, as in EMA and it updated significantly the information about the velum. The increase of temporal resolution, updates the quality of the analysis and allows the description of a wider set of sounds, including faster sounds as thrills and more complex sounds as diphthongs and nasal vowels. The significant increase of the number of participants is crucial to take apart individual from languages specific characteristics. The state-of-the-art as summarized in [2] includes currently (1) frame rates that already surpass 100 Hz; (2) databases for several languages were recorded at high frame rates (English [4], German [5], French [5]); (3) use of a wide variety of RT-MRI analysis techniques, that can be classified in four classes [2]; (4) basis decomposition or matrix factorization techniques at the level of the raw or processed images, pixel or region-of-interest (ROI) based, grid-based, and contour-based; (4) initial studies regarding the dynamics of articulators and gestural timing relationships.

Despite all these evolutions, particularly in extraction of information from the images (e.g., [5, 6]), the number of studies going beyond the example of contour or articulators is very scarce. A representative example of working in the facilitation of high level analysis is [7]. As frame rates rise, more work is needed in these frameworks for quantitative systematic analysis, that will be essential to make possible exploring the highly increased amount of images.

Several studies used MRI for studying EP, as summarized in Section 2. Compared to the state-of-the-art [2], these studies used a low frame rate, possibly providing not enough information to the adequate characterization of the investigated sounds. Further limitations are: (1) the scarce amount of publications and missing analysis of some sounds, namely rhotics and diphthongs; (2) preliminary character of the approaches to other sounds such as nasal vowels, due to the limited frame rate used and the amount of participants.

The main objectives of this paper are the following: (1) adding Portuguese to the small set of languages studied using the last evolutions in RT-MRI; (2) contribute to a better understanding of the dynamical aspects in the production of speech sounds; (3) assess previous results obtained with lower temporal resolution; (4) profit from the improved temporal resolution to start covering sounds characterized by their dynamic nature which could not be contemplated in previous studies, such as the diphthongs.

Even if essential for these objectives, the acquisition of
the novel data for new classes of sounds and more complex phonetic contexts with higher temporal resolution poses several challenges on how to process and analyze the resulting database. Our team has previously addressed this challenge, for different data [8, 9] by proposing methods to process [6] and analyze [7, 10] the image sequences to extract articulatory data. However, the nature of the novel database raises several new questions, opportunities and challenges, including the applicability of previous analyses.

The remainder of this article is organized as follows: Section 2 provides a short summary of related work in speech production studies using RT-MRI and other techniques, focusing in studies addressing EP. Section 3 presents information regarding a novel RT-MRI database acquired for EP, from corpus description to the methods considered for acquisition, processing and analysis of the image sequences; in Section 4, illustrative results are presented, including novel information regarding diphthongs production; finally, the article concludes with a brief summary of the contributions and ideas for future work.

2. Related Work

Nowadays, speech production studies can be supported by a wide range of technologies including imaging modalities (e.g., Ultrasound, MRI) and other instrumental techniques (e.g., EMA [11, 12]). In this context, real-time magnetic resonance imaging has been received particular relevance due to its non-invasiveness, non-use of ionizing radiation, and the remarkable improvements in spatio-temporal resolution achieved in the last few years [13, 2].

From the first attempts to acquire real-time imaging with MRI to date, much has been evolved in the area, achieving sampling rates close to those obtained with EMA and Ultrasongraphy. This has been possible by exploiting several technological advancements that involve the use of high field strengths, more powerful gradients, dedicated coils, non-Cartesian K-space trajectories, high degree of undersampled data and more efficient image reconstruction algorithms allowing for high sampling rates and improved image quality [5, 14].

For European Portuguese (EP), several of these techniques (e.g., EMA and MRI) have been used. These efforts include not only data acquisition but also data analysis. Early studies addressed dynamic aspects of nasal vowel production using EMA [15, 16, 17] and MRI (e.g. [9]). Internationally, several research groups have been using MRI to gather information for different languages using different approaches. Comprehensive reviews of these studies are summarized in [13, 2, 7].

The first MRI study for EP included 2D and 3D data regarding static configuration of all EP vowels, nasal consonants for one speaker [9]. A deeper study on EP laterals (3D) was conducted later with data from 7 participants [8, 18]. A study of co-articulation resistance in EP was presented in [19]. First results of RT-MRI for EP were presented in 2012 [3]. At this stage, a RT-MRI dataset which included nasals, velars, taps and trills was acquired with a frame rate of 14 fps. This represented an important first step towards a better characterization of the dynamic aspects involved in the production of these sounds of Portuguese [3]. The configuration of the vocal tract during the production of nasal vs. oral vowels was investigated using RT-MRI in [20]. More recently, RT-MRI was used for studying the temporal coordination of oral articulators and the velum during the production of nasal vowels [21], providing additional support for the delayed coordination of oral and nasal gestures in Portuguese [22, 21].

Despite all these relevant contributions, the sounds of Portuguese are not yet well described: diphthongs and trills still missing, nasal vowels need an improved characterization due to the lower frame rate and the small amount of participants.

From the perspective of processing and quantitative information extraction a lot has been achieved recently for EP exploring, essentially, automatic techniques to deal with the quantity and diversity of the information obtained, both from 3D static [18] and real-time MRI [20, 6, 7, 10]. However, the problem of information extraction for speech production studies and development of relevant articulation models is far from solved. The consideration of these imaging technologies poses challenges to extract articulatory-relevant information profiting from the full range of available data.

3. Methods

Gathering novel insights on the dynamic nature of complex sounds, for EP, taking advantage of recent advances in real-time MRI, involves several steps from corpus definition to articulatory analysis, as described in what follows.

3.1. Corpus

The corpus consists of minimal pairs containing all stressed oral [i, e, r, a, ɔ, o, u] and nasal vowels [æ, i, ə, ɔ, ʊ] in one and two syllable words. Nasal diphthongs /aw, ej, ij/ and the oral counterparts /aw, aj, oj/ as well as /ɛj, ɛw, ow, ʊj/ in monosyllabic words were also included. Additional materials were recorded for further modeling of variability in the production of nasality.

All words were randomized and repeated in two prosodic conditions embedded in one of three carrier sentences alternating the verb as follows (Diga ‘Say’—ouvi ‘I heard’—leio ‘I read’) as in ‘Diga pote, diga pote baixinho’ (‘Say pot, Say pot gently’). So far, this corpus has been recorded from twelve native speakers (8m, 4f) of EP. The tokens were presented from a timed slide presentation with blocks of 13 stimuli each. The single stimulus could be seen for 3 seconds and there was a pause of about 60 seconds after each block of 13. The first three participants read 7 blocks in a total of 91 stimuli and the remaining nine participants had 9 blocks of 13 stimuli (total of 117 tokens).

3.2. RT-MRI Acquisition

RT-MRI recordings, as exemplified in Fig. 1, were conducted at the Max Plank Institute for biophysical Chemistry, Göttingen, Germany, using a 3 Tesla Siemens Magnetom Prisma Fit MRI System equipped with high performance gradients (Max amplitude 80 mT/m; slew rate = 200 T/m/s).

A standard 64–channel head coil was used with a mirror mounted on top of the coil. The speaker was lying down, in a comfortable position, and was instructed to read the required sentences. Real-time MRI measurements were based on a recently developed method, where highly under-sampled radial FLASH acquisitions are combined with nonlinear inverse reconstruction (NLINEV) providing images at high spatial and temporal resolutions [23]. Acquisitions were made at 50 fps, resulting in images as the ones presented in Fig. 1. Speech was synchronously recorded using an optical microphone (Dual Channel-FOMRI, Optoacoustics, Or Yehuda, Israel), fixed on the head coil, with the protective pop-screen placed directly against the speaker’s mouth.
Before their enrollment on the study, all volunteers provided informed written consent and filled an MRI screening form in agreement with institutional rules. None of the participants had any known language, speech or hearing problems and were compensated for their participation.

3.3. Speech annotation

The speech recordings were preprocessed to filter MRI acquisition noise and the target segments manually delimited by the first author using Praat [24, 25]. The produced annotation information was used in Matlab for extraction of relevant images and synchronized speech signal.

3.4. Processing and Analysis

In line with the semi-automated methods described in Silva et al. [7], twenty-eight images were manually annotated to train two active appearance models [26] for oral and nasal configurations of the vocal tract. For the work presented in this article, the RT-MRI sequences for one of the speakers (male, 39 yo) were processed to extract sequences of vocal tract contours. Figure 1 illustrates the overall outcome of the processing stage by presenting a few selected image frames from the sentence 'Diga vu' (say vu) depicting the identified vocal tract contours, with the different articulators shown in different colors and line types, and the corresponding audio annotation.

The static and dynamic analysis and comparison among vocal tract configurations was performed by adopting the framework proposed in Silva et al. [10] providing objective normalized analysis and visualization. Accordingly, vocal tract configurations are compared for seven different regions: velum (VEL), tongue dorsum (TD), tongue back (TB), tongue tip (TT), lip protrusion (LP), lip aperture (LA) and pharynx (Ph). For each, the comparison yields a score, from 1 (no difference) to 0 (strong difference), which is represented over the unitary circle (Fig. 2) or graph (Fig. 3) for static and dynamic analysis, respectively. For the sake of brevity, the reader is forwarded to [10] for additional details regarding the adopted analysis framework.

4. Results

Considering our initial goals, to start exploring the new corpus and assess the applicability of the previously proposed processing and analysis methods to this new context, we processed the data for one of the male speakers. Our aim was to explore two important aspects: confirm overall previous findings, as a proof of replicability, particularly for oral and nasal vowels, and obtain a first insight over EP diphthongs.

As an example, Fig. 2 shows the comparison between [a] and [i] performed for the previous 14 fps corpus and using data from the new 50 fps corpus. The visual representation, in Fig. 2, should be interpreted as the answer to the question: "What happens when I move from [a] to [i]?". As can be observed, the difference polygon, inscribed in the unitary circle, has grossly the same shape, for both corpora, and depicts the same notable differences, namely (dots in yellow and red circular coronas): the tongue dorsum moving up, and an advance of the tongue tip and tongue back. Overall, the similarity between the diagrams (results) obtained for both corpora corroborates the reproducibility of the analysis methods among datasets.

For the dynamic analysis of nasal vowels, Fig. 3 shows, as an example, the evolution of the vocal tract configuration along the production of [e]: it is essentially characterized by velum movement (noticeable by a trajectory entering in the yellow zone of the graph at around 30%). Of note is, also, the variation for the tongue dorsum and tongue tip, observed for the
end of the vowel and a possible consequence of coarticulation effects. For instance, the influence of the carrier sentence’s second ‘diga’ (say), following the token, as in ‘Diga vem, diga . . . ‘ (Say come, say . . . ).

4.1. EP Diphthongs

Given the exploratory nature of this first effort regarding diphthongs, to have a first grasp of what is happening, we started by oral diphthongs. As an example, Fig. 4 shows results for [aw]

![Figure 4: Variation, over time, of the articulators during the production of the EP oral diphthong [aw], as in 'pau' (stick).](image)

as in ‘pau’ (stick). The diphthong is covered by 10 images (10 points in the graph), including the initial [p] and the diphthong (around 200 ms). Note that these representations only present lines, in the graph, for those regions where, along production, changes fall into the yellow and red stripes. It is clear an abrupt change in lip aperture (LA) at 20-30 % of diphthong duration, from [p] to [a], and a reduction, close to the end, for the [w].

The nasal counterpart of this diphthong, [˜iw], as in ‘pão’ (bread), is analyzed in Fig. 5, showing a gradual variation of lip aperture (LA), similar to the one observed in paw. Additionally, there is movement of the tongue back, as opposed to the oral counterpart, probably due to the need of adjustment in the nasal passage, which is hinted by the changes also noted at the velum.

![Figure 5: Variation, over time, of the articulators during the production of the EP oral diphthong [˜iw], as in 'pão' (bread).](image)

Our corpus and methods also enable investigating the complex context of a diphthong after a nasal consonant, and, for instance, in comparison with other diphthongs. In Fig. 6, we compare the production of ‘mão’ ([miw], ‘hand’) with ‘pão’ ([piw], ‘bread’), showing that, as expected, the velum behaves differently, in the initial part, since it is open, from the beginning, in ‘mão’.

5. Conclusion

The major contribution of this paper is the presentation of a novel RT-MRI database for EP recorded with a frame rate of 50 Hz, contributing to augment the very reduced amount of languages with such a valuable resource for speech production studies. This database, after its completion and pre-processing will be partially made available for other researchers. Additionally, we demonstrate the applicability of previously proposed methods for segmentation and analysis, by illustrating previous findings for oral and nasal vowels and performing a first exploration of EP diphthongs. The application of this methodology to more data and speakers will enable a detailed description of nasal sounds in European Portuguese and a better understanding of their implementation in production.

The work presented here can still profit from several improvements and provides the grounds for exploring new routes of speech production studies in EP. Even though the image quality is better than our previous 14 fps corpus, the different nature of the corpus, with a large number of dental, alveolar and palatal contacts (in the support words and sentences, e.g., [t] before [iw] as in ‘sotão’ – attic) poses new challenges to vocal tract segmentation, with a few segmentations still requiring a final manual revision, an aspect to improve as new speakers are included, by training better models [6].

Finally, now that the grounds for work have been established, future developments should be propelled by addressing concrete hypotheses regarding EP nasals, such as the one of delayed coordination of oral and nasal gestures in Portuguese [27, 21, 22].

6. Acknowledgements

This work is partially funded by the project ‘Synchrone Variabilität und Lautwandel im Europäischen Portugiesisch’, with funds from the German Federal Ministry of Education and Research, by IEETA Research Unit funding (UID/CEC-00127/2013), by Portugal 2020 under the Competitiveness and Internationalization Operational Program, and the European Regional Development Fund through project SOCA – Smart Open Campus (CENTRO-01-0145-FEDER-000010) and project MEMNON (POCI-01-0145-FEDER-028976). We thank Philip Hoole for the scripts for noise suppression and all the participants of the experiment for their time and voice.
7. References


A postfiltering approach for dual-microphone smartphones

Juan M. Martín-Doñas\textsuperscript{1}, Iván López-Espejo\textsuperscript{2}, Angel M. Gómez\textsuperscript{1}, Antonio M. Peinado\textsuperscript{1}

\textsuperscript{1}Dept. de Teoría de la Señal, Telemática y Comunicaciones, Universidad de Granada, Spain
\textsuperscript{2}VeriDas | das-Nano, Spain

\{mdjuanart,amgg,amp\}@ugr.es, ilopez@das-nano.com

Abstract

Although beamforming is a powerful tool for microphone array speech enhancement, its performance with small arrays, such as the case of a dual-microphone smartphone, is quite limited. The goal of this paper is to study different postfiltering approaches that allow for further noise reduction. These postfilters are applied to our previously proposed extended Kalman filter framework for relative transfer function estimation in the context of minimum variance distortionless response beamforming. We study two different postfilters based on Wiener filtering and non-linear estimation of the speech amplitude. We also propose several estimators of the clean speech power spectral density which exploit the speaker position with respect to the device. The proposals are evaluated when applying speech enhancement on a dual-microphone smartphone in different noisy acoustic environments, in terms of both perceptual quality and speech intelligibility. Experimental results show that our proposals achieve further noise reduction in comparison with other related approaches from the literature.

Index Terms: Postfiltering, Extended Kalman filter, Minimum variance distortionless response, Dual-microphone speech, Smartphone

1. Introduction

Multi-channel speech processing is widely employed in devices with several microphones to improve the noise reduction performance, yielding better speech quality and/or intelligibility. The most common techniques for multi-channel processing are the beamforming algorithms, which apply a spatial filtering to the existing sound field [1, 2, 3]. Nevertheless, the performance of beamforming can be insufficient if a small number of microphones is considered, as in the case of dual-microphone smartphones [4, 5]. Thus, to improve the noise reduction performance, the enhanced signal at the output of the beamformer might be further processed by using postfiltering methods [6].

Several postfilters have been proposed in the recent years mainly based on multi-channel Wiener filtering, which can be decomposed into minimum variance distortionless response (MVDR) beamforming plus single-channel Wiener filtering [3]. For multi-channel Wiener filtering, Zelinski [7] assumed a spatially-white noise to estimate the speech and noise statistics. Marro \textit{et al.} [8] further improved the postfilter architecture and also considered acoustic echo and reverberation. The previous spatially-white noise assumption was substituted in [9] by assuming a diffuse noise field. The postfilter presented in [10] took into account the noise reduction provided by the beamformer to obtain more accurate statistics. That work also explores non-linear postfilters based on minimum mean square error (MMSE) estimation of the speech amplitude. Gannot \textit{et al.} [11] modified the beamformer using a generalized sidelobe canceler (GSC) structure and an optimal modified log-spectral amplitude (OMLSA) estimator [12] as a postfilter. Apart from the above, a statistical analysis of dual-channel postfilters in isotropic noise fields is presented in [13].

In the case of dual-microphone smartphones, other works have exploited the information of the secondary microphone to enhance the speech signal from the reference microphone. For example, Jeub \textit{et al.} [14] proposed an estimator of the noise power spectral density (PSD) along with a modified single-channel Wiener filter that explicitly exploits the power level difference (PLD) of the speech signal between the microphones in close-talk (CT) conditions (when the loudspeaker of the smartphone is placed at the ear of the user). Nelke \textit{et al.} [15] developed an alternative noise PSD estimator in far-talk (FT) conditions (when the user holds the device at a distance from her/his face). This method combines a single-channel speech presence probability estimator and the coherence properties of the dual-channel target signal and background noise. The noise PSD is employed to estimate the gain function to be applied on the reference channel. Such an algorithm was extended for multi-microphone devices in [16]. Nevertheless, all of these techniques make assumptions about the noise field properties that may not be accurate in practice, thereby leading to a limited performance.

Recently, we proposed an estimator of the relative transfer function (RTF) between microphones based on an extended Kalman filter (eKF) framework [17]. Our method is capable of tracking the RTF evolution using prior information on the channel and noise statistics without making any assumption on the clean speech signal statistics. In that work we evaluated the performance of our estimator over MVDR beamforming applied on a dual-microphone smartphone configuration in CT and FT conditions, showing improvements in estimation accuracy with respect to other relevant methods from the literature. Despite this, the speech enhancement performance is still limited as a result of using beamforming with only two microphones.

In this paper we evaluate the use of postfiltering techniques to overcome shortcomings of our eKF-based MVDR approach for dual-microphone smartphones. We compare different algorithms and modify them to adapt them to our eKF-based method. Also, we propose different clean speech PSD estimators and make use of the available information about the RTF and noise to obtain the needed statistics for postfiltering. Our proposals are evaluated on a dual-microphone smartphone under several noisy acoustic environments in CT and FT condi-
tions, achieving improvements in terms of noise reduction performance in comparison with other state-of-the-art approaches.

The remainder of this paper is organized as follows. In Section 2 we briefly revisit the eKF-based RTF estimation and its application to MVDR beamforming. Section 3 describes the proposed postfiltering approaches and clean speech PSD estimators for CT and FT conditions. Then, in Section 4 the experimental framework is presented along with our perceptual quality and speech intelligibility results. Finally, conclusions are summarized in Section 5.

2. Beamforming for dual-microphone smartphones

Before presenting the postfiltering approaches, it is worthwhile to review the RTF estimation and beamforming for dual-microphone smartphones that we proposed in [17]. First, we introduce formulation proposed for the eKF-based RTF estimation. Next, we describe the MVDR beamforming approach for processing the dual-channel noisy speech signals using the noise statistics and RTF estimations.

2.1. Extended Kalman filter-based RTF estimation

Let us consider the following additive distortion model for the noisy speech signal in the short-time Fourier transform (STFT) domain,

\[ y_m(f, t) = x_m(f, t) + n_m(f, t), \]

where \( y_m(f, t), x_m(f, t) \) and \( n_m(f, t) \) represent, respectively, noisy speech, clean speech and noise STFT coefficients at the \( m \)-th microphone (\( m = 1, 2 \)), \( f \) is the frequency bin and \( t \) the frame index. Using the relative transfer function (RTF) \( A_2(f, t) = \frac{Y_2(f, t)}{X_2(f, t)} \) between both microphones, we can write the speech distortion model for the secondary microphone (\( m = 2 \)) in terms of the reference microphone (\( m = 1 \)) as

\[ Y_2(f, t) = A_2(f, t) (Y_1(f, t) - N_1(f, t)) + N_2(f, t). \]

We can also rewrite the previous complex variables as vectors stacking their real and imaginary parts, yielding \( y_1^{(i)}, a_1^{(i)} \) and \( n_1^{(i)} \) (index \( f \) is omitted for clarity). For example, we define the noisy speech vector for the \( m \)-th microphone as

\[ y_1^{(i)} = \begin{bmatrix} \text{Re}(y_m(t)) \; \text{Im}(y_m(t)) \end{bmatrix}^T, \]

where \([\cdot]^T\) indicates transpose. Then, we set a dynamic model for \( a_1^{(i)} \) as follows,

\[ a_1^{(i)} = a_1^{(i-1)} + w^{(i)}, \]

where \( w^{(i)} \) models the variability of the RTF between consecutive frames. Also, we redefine (2), using the previous vector notation, as

\[ y_2^{(i)} = \begin{bmatrix} C \; D \end{bmatrix} \begin{bmatrix} y_1^{(i)} - n_1^{(i)} \end{bmatrix} + \begin{bmatrix} a_1^{(i)} \; n_2^{(i)} \end{bmatrix}, \]

where \( C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \) and \( D = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \).

Assuming multivariate Gaussian variables and using the models in (4) and (5), in [17] we proposed an MMSE estimator of \( a_1^{(i)} \) using an extended Kalman filter (eKF) framework, which tracks the RTF using the observable noisy speech, estimated noise statistics and a priori information on the RTF statistics [17].

2.2. MVDR beamforming

Once the RTF is estimated, both noisy signals are combined using MVDR beamforming, whose weights can be expressed (omitting indices \( f \) and \( t \) for clarity) as [3]

\[ F = \frac{\Phi_{nn}^{-1}d}{d^H\Phi_{nn}^{-1}d}, \]

where \((\cdot)^H\) indicates Hermitian transpose, \( d = [1, \; A_2]^T \) is the steering vector and \( \Phi_{nn} \) is a noise spatial correlation matrix (i.e., obtained from \( \Phi = [N_1, \; N_2]^T \)). We can also express the PSD of the noise at the output of the beamformer as [10]

\[ \phi_n = (d^H\Phi_{nn}^{-1}d)^{-1}. \]

Finally, the enhanced signal for the reference microphone is estimated as \( \hat{X}_{1,\text{mvdr}} = FY \), with \( Y = [Y_1, \; Y_2]^T \).

3. Postfiltering for dual-microphone smartphones

The performance of MVDR beamforming when applied on smartphones with only two microphones is quite limited due to the reduced spatial information and the particular placement of the microphones [4]. Therefore, we analyze the use of postfiltering for enhancing the signal at the output of the beamformer and further improve the noise reduction performance. We propose two different postfilters based on Wiener filtering and the optimal modified log-spectral amplitude (OMLSA) estimator [12], and also address the estimation of the clean speech PSD at the reference microphone needed by the postfilters. In addition, the postfiltering gains are further processed using the musical noise reduction algorithm proposed in [18]. This post-processing is applied to frequencies above 1 kHz. For ease of notation, we drop the indices \( f \) and \( t \) henceforth.

3.1. Wiener filtering

The multi-channel Wiener filter can be decomposed into an MVDR beamformer followed by a single-channel Wiener filter defined as

\[ G_{\text{wt}} = \frac{\xi}{1 + \xi}, \]

where \( \xi = \phi_{x_1}/\phi_n \) is the a priori signal-to-noise ratio (SNR) and \( \phi_{x_1} \) is the clean speech PSD at the reference microphone. This Wiener filter is partially obtained from the enhanced signal at the output of the beamformer. Better performance can be achieved if the Wiener filter is fully calculated from the reference noisy signal when an overestimated noise is considered. Thus, we propose the following improved Wiener filter

\[ G_{\text{wt}} = \frac{\hat{\phi}_{x_1}}{\phi_{x_1} + \mu\phi_n}, \]

where \( \hat{\phi}_{x_1} \) is an estimate of the clean speech PSD (discussed in Subsection 3.3), \( \phi_n \) is an estimate of the noise PSD, taken as \( \phi_{n_1} \) (first element of the diagonal of \( \Phi_{nn} \)) in order to use an overestimated version of the noise, and \( \mu \) is a factor which provides an increased overestimation. As a result, the clean speech signal is estimated as \( \hat{X}_{1,\text{wt}} = G_{\text{wt}}X_{1,\text{mvdr}} \).
3.2. Optimal modified log-spectral amplitude estimator

The OMLSA estimator proposed in [12] computes the postfilter gains as

\[ G_{\text{omlsa}} = G_{\text{HI}}^{\text{new}} G_{\text{HI}}^{1-p_{\text{SSP}}}, \tag{10} \]

where \( p_{\text{SSP}} \) is the speech presence probability, \( G_{\text{HI}} \) is a constant gain when speech is absent and \( G_{\text{af}} \) is the gain when speech is present, computed as

\[ G_{\text{HI}} = G_{\text{af}} \exp \left( \frac{1}{2} \int_{-\infty}^{\infty} \frac{G_{\phi,u}}{\gamma} dt \right). \tag{11} \]

where \( \gamma = \sqrt{X_{1, \text{omlsa}}}^2/\phi_u \) is the a posteriori SNR and \( G_{\text{af}} \) was defined in (8). We modify this gain by substituting \( G_{\text{af}} \) by \( G_{\text{af}} \), defined in (9), which yields the improved OMLSA gain \( G_{\text{imlsa}} \). Finally, the clean speech signal is estimated as

\[ \hat{X}_{1, \text{omlsa}} = G_{\text{omlsa}} \hat{X}_{1, \text{mvdr}}. \]

3.3. Clean speech PSD estimators

The previous postfilters require an estimation of the clean speech PSD, \( \phi_{x_1} \). We propose two different estimators for CT and FT conditions, respectively, based on the noisy speech and noise statistics and the estimated RTF between microphones. Therefore, these estimators take advantage of the more accurate RTFs obtained by our eKF-based approach.

For close-talk (CT) conditions, the estimator is based on the PLD between microphones [14], which can be computed as

\[ \Delta \hat{\phi}_{\text{PLD}} = \max (\hat{\phi}_{x_1} - \hat{\phi}_{x_2}, 0), \tag{12} \]

where \( \hat{\phi}_{x_1} \) and \( \hat{\phi}_{x_2} \) are the noisy speech PSDs at the reference and secondary microphones, respectively. This estimator takes advantage of the more accurate clean speech component at the secondary microphone. Assuming that the noise PSD is similar at both microphones so that its difference can be neglected compared to \( \Delta \hat{\phi}_{\text{PLD}} \), it can be easily shown that the clean speech PSD can be approximated as [14]

\[ \hat{\phi}_{x_1}^{\text{CT}} = \Delta \hat{\phi}_{\text{PLD}} / (1 - |A_{21}|^2). \tag{13} \]

Unlike CT conditions, in far-talk (FT) conditions speech power is similar at both microphones and the previous assumptions are inaccurate (i.e., noise PSD difference cannot be neglected compared to PLD between microphones) [14]. Therefore, a better estimator is obtained by considering the distortionless properties of MVDR beamforming, which imply that the clean speech PSD at the reference microphone is the same as the one at the beamformer output. Thus, we estimate the clean speech PSD at both the beamformer output and the reference microphone as

\[ \hat{\phi}_{x_1}^{\text{FT}} = \Phi_{y y}^{-1} (\Phi_{y n} - \Phi_{n n}) F, \tag{14} \]

where \( \Phi_{y y} \) is a noisy speech spatial correlation matrix (i.e., calculated from \( Y \)), whose diagonal elements are the noisy speech PSDs \( \phi_{y_1} \) and \( \phi_{y_2} \). Although the clean speech estimate could be obtained from a direct subtraction of the first diagonal elements of matrices \( \Phi_{y y} \) and \( \Phi_{y n} \), the estimator defined in (14) has the advantage of using all the channels in the estimation.

4. Experimental evaluation

4.1. Experimental framework

To evaluate the proposed techniques, we simulated dual-channel noisy speech recordings on a Motorola Moto G smartphone. We consider two different modes of use: close-talk (CT) and far-talk (FT). These modes can be easily identified using the proximity sensor included in the smartphone. Clean speech signals were obtained from 18 speakers of the VCTK database [19] downsampled to 16 kHz. We simulated recordings at eight different noisy environments with different reverberations: car, street, babble, mall, bus, cafe, pedestrian street and bus station. The noise signals were added at six different SNR levels from -5 dB to 20 dB. Further details about this database can be found in [17].

For STFT computation, we choose a 25 ms square-root Hann window with 75% overlap. The noisy speech spatial correlation matrix \( \Phi_{\phi_1} \) is estimated by a first-order recursive averaging with an averaging constant of 0.9. The noise spatial correlation matrix \( \Phi_{\phi_n} \) is estimated by recursive averaging during time-frequency bins where speech is absent. Thus, we compute the speech presence probability \( p_{\text{SSP}} \) at each bin by using the Multi-Channel Speech Presence Probability (MC-SPP) noise tracking algorithm proposed in [20]. Finally, we use an overestimation factor \( \eta = 4 \) and a speech absent gain \( G_{\text{hi}} = 0.05 \) for postfiltering implementation.

4.2. Results

The two proposed postfilters, Wiener filtering (eKF-WF) and OMLSA estimator (eKF-OMLSA), are evaluated in combination with MVDR beamforming, both using the eKF-based RTF estimator outlined in Subsection 2.1. The obtained results are compared with those achieved by the noisy speech at the reference microphone, and MVDR beamforming with eKF and no postfiltering (eKF-MVDR) [17]. Also, we evaluate two other state-of-the-art enhancement algorithms for dual-microphone smartphones, that is, the PLD-based Wiener filtering for close-talk conditions of [14] and the speech presence probability and coherence-based (SPPC) Wiener filtering for far-talk conditions of [15]. The musical noise suppressor of [18] is also applied to PLD and SPPC gains.

The resulting enhanced signals are assessed in terms of perceptual quality and speech intelligibility by means of the perceptual evaluation of the speech quality (PESQ) [21] and short-time objective intelligibility (STOI) [22] metrics, respectively. Clean speech at the reference microphone is taken as a reference for these performance metrics. The results for close-talk (CT) and far-talk (FT) conditions are shown in Tables 1 and 2, respectively.

In close-talk conditions, it is shown that the proposed postfilters outperform the other methods in terms of perceptual quality, with both Wiener filtering and OMLSA approaches obtaining similar performance on average. While the Wiener filtering approach achieves better PESQ results at low and medium SNRs, the OMLSA approach yields higher PESQ scores at high SNRs. It can also be seen that PLD is a better choice than eKF-MVDR, but the addition of postfiltering after beamforming leads to better PESQ results. On the other hand, intelligibility scores among the different evaluated techniques are similar, but MVDR beamforming without postfiltering obtains slightly better ones. That means that the superior perceptual quality achieved by postfiltering involves some speech distortion that slightly reduces intelligibility. In general, PLD and Wiener
postfiltering have a similar performance, while OMLSA shows slightly worse results, especially at low SNRs. Thus, eKF-WF seems the preferred strategy for close-talk conditions on average.

Regarding far-talk conditions, likewise, the proposed postfilters obtain the best results in terms of perceptual quality, with Wiener filtering being the best strategy for noise reduction, especially at high SNRs. The SPPC method outperforms MVDR beamforming with no postfiltering, but it does not achieve any improvements compared to eKF-WF and eKF-OMLSA. However, SPPC introduces more speech distortion, yielding a poor performance in terms of speech intelligibility. MVDR beamforming with no postfiltering achieves the best STOI scores, as in CT conditions. On the other hand, the postfiltering approaches obtain similar results on average, although their performance is worse at low SNRs. The comparison of both postfilters indicates that eKF-OMLSA achieves better intelligibility on average and, especially, at low SNRs. To sum up, both eKF-WF and eKF-OMLSA perform similarly, with Wiener filtering achieving best perceptual quality and OMLSA better speech intelligibility in FT conditions.

### 5. Conclusions

In this paper we have proposed a postfiltering approach to our RTF extended Kalman filter framework for dual-microphone smartphones. Our proposals make use of the more accurate estimated RTFs and noise statistics in order to obtain the gain function for noise reduction. We evaluated two different postfilters based on Wiener filtering and the OMLSA estimator, and also proposed different clean speech PSD estimators for CT and FT conditions in order to compute the needed statistics. The proposed approaches were evaluated in terms of perceptual quality and speech intelligibility when they are used for enhancing noisy speech signals from a dual-microphone smartphone in adverse acoustic environments. Our results show improvements in terms of both perceptual quality and noise reduction of the enhanced signal while low speech distortion is introduced in comparison to a standalone MVDR beamformer. As future work, we will extend this study on postfiltering to general multi-microphone devices through our extended Kalman filter approach.

### 6. References


Speech and monophonic singing segmentation using pitch parameters

Xabier Sarasola, Eva Navas, David Tavarez, Luis Serrano, Ibon Saratxaga
AHOLAB University of the Basque Country (UPV/EHU), Bilbao, Spain
{xsarasola,eva,david,lserrano,ibon}@aholab.ehu.eus

Abstract

In this paper we present a novel method for automatic segmentation of speech and monophonic singing voice based on two parameters derived from pitch: proportion of voiced segments and percentage of pitch labelled as a musical note. First, voice is located in audio files using a GMM-HMM based VAD and pitch is calculated. Using the pitch curve, automatic musical note labelling is made applying stable value sequence search. Then pitch features extracted from each voice island are classified with Support Vector Machines. Our corpus consists in recordings of live sung poetry sessions where audio files contain both singing and speech voices. The proposed system has been compared with other speech/singing discrimination systems with good results.

Index Terms: audio segmentation, voice discrimination, singing voice, pitch

1. Introduction

There have been many studies dealing with the analysis and development of technologies for speech. Lately there is also an interest in studying singing speech. Both areas are intimately related, but it is not possible to apply directly the same techniques to both problems. For instance, phone alignment systems with good performance for speech do not get good results for singing voice [1]. Also automatic recognition systems degrade their performance when dealing with singing speech [2]. Despite this different behaviour with speech technologies both types of voice share the same nature and therefore are very close. In fact, singing voice is considered an extension of common speech and sometimes it is not easy to distinguish them. Even for humans it is sometimes difficult to correctly discriminate speech from singing voices [3], [4]. Nevertheless, there is an agreement about some characteristics that behave differently between speech and singing [5], [6]:

- Voiced/unvoiced ratio is higher in singing voice, as vowels tend to be longer to accommodate to the duration of notes.
- Dynamic ranges of energy and F0 in singing voice are higher than in speech.
- Vibrato areas can appear in long sustained notes in singing voice. This phenomenon never occurs in speech.
- The pitch continuity is different in speech and singing voice. Pitch in singing voice is more similar to a series of discrete values and on the contrary in speech it is a continuously varying curve.

Two types of automatic speech/singing classification systems have been proposed. The first scheme consists on using short-time features of the signal, like for instance spectral information in the form of MFCCs, to classify it at frame level [7]. In [8], a wide range of short-time features are analysed for frame level classification of polyphonic music, singing voice and speech. The second approach uses long-term features for classification. In previous works, distribution of pitch values [9], pitch parameters [10] and note grammars [11], [12] are used. These long term features can be calculated for the whole file or for a window that slides through the audio file. The former case corresponds to speech/singing classification and the latter to segmentation. For the segmentation problem, the results improve using long windows up to 1000 ms [13].

Bertsolarita is a live improvisation poetry art from Basque Country. In these live sessions a host introduces the singer and proposes the topic for the improvised verses that will be sung a capella. The singers are professional verse creators, but not professional singers. Many live sessions of bertsolarita are recorded and saved together with the corresponding transcriptions by Bertso associations. All these recordings are very useful for analysis of the singing style and as training database for bertso synthesis. Each recording includes speech from the host, sung verses and overlapped applauses. Thus, a good segmentation system is required to isolate the segments of interest.

Our goal is to create a tool to automatically segment the bertsolarita audio files to get the singing voice.

The rest of the paper is organized as follows. Section 2 explains the proposed segmentation system using only two parameters derived from pitch. Section 3 details the algorithm developed to assign a musical note label to each audio frame. Section 4 describes the results of the segmentation system and compares it to other speech/singing voice classification systems. Finally, Section 5 presents the conclusions of the work.

2. Proposed segmentation system

The main scheme of the proposed system can be seen in Figure 1. First voice is detected in the audio files using a voice activity detection (VAD) algorithm based on Hidden Markov Models and Gaussian Mixture Models (GMM-HMM). Pitch contour is calculated and for each voiced segment pitch parameters are extracted. These parameters are classified as speech or singing by a previously trained Support Vector Machine (SVM).

2.1. GMM-HMM based VAD

The first stage of the proposed system divides the audio in three classes: voice, noise and silence. For this purpose we have used a GMM frame-level classifier smoothed with an HMM as used in [14]. We have considered 13 MFCC values and $\Delta$ and $\Delta^2$ values with 25 ms window and 10 ms frame period. GMMs are trained using Expectation Maximization (EM) for speech, noise and silence. Frame-level classification usually creates noisy segmentation because in the transition segments fast label changes are produced. To avoid this, we have used an ergodic HMM of three states, one per class. This HMM has predefined transition probabilities with preference for remaining in the current state. In our case, the transition probability of the HMM
outside the diagonal is 0.0001. This way, fast transitions and small discontinuities are removed. To classify the segments, the likelihood of observation provided by each model is calculated using expression (1).

\[
P(o|s_i) = \sum_{j=1}^{M} w_{ij} N(o|\mu_{ij}, \Sigma_{ij})
\]

where \( o \) is the MFCC vector, \( w_{ij}, \mu_{ij}, \Sigma_{ij} \) are the weight, mean and diagonal covariance of the component \( j \) of the state \( s_i \) and \( M \) is the number of Gaussian components.

### 2.2. Singing/speech classification

We have considered each audio segment labelled as voice by the GMM-HMM VAD as corresponding to a unique class: either speech or singing. We do not consider the option of having speech and singing in the same voice island because of the characteristics of our database. To evaluate a system that includes speech and singing in the same voice island because of the characteristics of our database. Therefore the problem is reduced to a binary classification of each voice island. We propose the use of two parameters derived from pitch to do the classification: proportion of voiced segments (PV) and percentage of pitch labelled as a musical note (PN). The pitch curve has been calculated using PRAAT autocorrelation method [15] with a frame period of 10 ms.

Voiced/unvoiced segments are obtained directly from the pitch curve and stable musical note segments are found using our algorithm explained in Section 3. The features for classification are calculated according to expressions (2) and (3).

\[
PV = \frac{N_{VF}}{N_T}
\]

where \( N_{VF} \) is the total number of voiced frames, \( N_{NF} \) is the total number of frames labelled as a musical note and \( N_T \) is the total number of frames, all of them calculated within the segment to be classified. The vector containing these two parameters is classified using a SVM.

### 3. Note detection algorithm

Our algorithm to label the pitch curve with musical notes discretises the F0 curve in semitones expressed in cents and then searches for sequences of semitones that fulfil two conditions: to have enough duration and less variation range than a threshold. This algorithm is simpler than state-of-the-art algorithms [16] but our lack of labelled data made us create a method with minimum supervision. First we map the F0 value to cent scale with an offset to make all the possible values of \( f_{0c} \) positive according to expression (4).

\[
f_{0c} = 1200 \log_{2}(\frac{f_{0}}{f_{ref}}) + 5800
\]

where \( f_{ref} \) is 440 Hz, the frequency of A4 note.

To avoid possible instability due to vibrato, we apply a smoothing to the F0 curve. The smoothing consists on calculating the local maxima and minima envelopes and taking the average curve. The obtained smoothed pitch curve is rounded to the closest semitone value to discretise the sequence, as shown in Figure 2.

### 3.1. SVM classifier

Using the semitone discretised pitch curve we search for notes using subsequence search techniques [17], [18].
search the non-overlapping group of subsequences that fulfil the predefined conditions of minimum length and maximum range in amplitude expressed in equations 5 and 6.

\[
\text{Len}(s) \geq L \quad (5)
\]

\[
\text{max}(s) - \text{min}(s) \leq R \quad (6)
\]

where \( s \) is the semitone subsequence, \( R \) is the maximum amplitude range and \( L \) is the minimum length.

The algorithm is defined in the next steps:

- Consider the whole pitch curve as a collection of \( K \) voiced sequences of contiguous semitones expressed in cents each one with its own length \( S = \{S_1, S_2, \ldots, S_K\} \).
- Define \( R \) as the maximum allowed variation range and \( L \) as the minimum length.
- Search the longest subsequence in each \( S_i \) (\( 1 \leq i \leq K \)) that fills the conditions.
- Label the longest subsequence found as a musical note. Between the possible semitones in the sequence, the most frequent one is selected as label.
- Split the remaining parts of the original sequence \( S_i \) in two new sequences: the subsequence in the left of the note found (\( S_{Li} \)) and the subsequence in the right of the note found (\( S_{Ri} \)) as shown in Figure 3.
- If any of the new generated subsequences fill the duration condition, \( S_i \) in \( S \) is substituted by them and the process begins again.
- When all sequences from \( S \) have been analysed the process finishes.

In this work we have established a maximum range of 100 cents and a minimum length of 150 ms, a standard minimum value in Western music [19].

### 4. Experiments and results

#### 4.1. Datasets

As few publicly available database exist with speech and monophonic singing, we have used an excerpt of our Bertsolaritza database [20] to train the algorithms and the NUS Sung and Spoken Lyrics Corpus [21] to test them. In the Bertso database we manually labelled 20 audio files from 19 singers with a total duration of 60 minutes and 40 seconds. These audio files contain 32.8 minutes of singing voice and 2.87 minutes of speech. The 20 files were selected to cover the variability of the original Bertsolaritza database, considering recordings from different decades and gender. In the NUS database, each singer has recorded a singing and spoken version of 4 songs. The total number of different songs is 20 and 12 singers have made the recordings, 6 males and 6 females. As each recording contains either speech or singing voice, we used the VAD to obtain the voice segments and labelled them with the type of the recording.

Table 1 shows the distribution of singers and hosts by gender in the Bertso database. In most cases the speakers either sing or act as host, but some hosts give the topic for the improvised verses singing as well. These hosts appear in the recordings both singing and speaking.

In the Bertso database the audio files originally were in mp3. Both databases had 44100 Hz samplerate and have been downsampled to 16000 Hz and converted to Windows PCM files.

We analysed the Bertso database and the singing voice and speech segments never appear consecutively, i. e., there is always silence or noise between segments to classify. Therefore, considering the structure of both databases, each segment belongs only to a class. Additionally, we have studied the duration of the segments produced by speakers and singers: singing voice has longer durations than speech (mean duration of 3.69 and 1.51 seconds respectively in Bertso database and 3.87 and 1.92 seconds in NUS database).

The distribution of the proposed classifying features \( PV \) and \( PN \) in the databases can be seen in Figures 4 and 5. Speech is more scattered than singing voice, but taking into account both parameters a good discrimination of both classes can be achieved.

For the experiments, we split the Bertso database in 10 subsets for cross-validation tests. All the partitions considered include different singers in the train and test subsets. The NUS database is classified using the algorithms trained with Bertso database.

<table>
<thead>
<tr>
<th>Singer</th>
<th>Host</th>
<th>Singer and host</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>7</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>Male</td>
<td>12</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>Total</td>
<td>19</td>
<td>12</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PV</th>
<th>Speech</th>
<th>Singing Voice</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.2</td>
<td>0.4</td>
</tr>
<tr>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>0.4</td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>0.6</td>
<td>0.8</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Figure 4: Distribution of the classes in the Bertso database

Figure 5: Distribution of the classes in NUS database

---

4.2. Comparison with other methods

To compare our algorithm with other methods we had tested them in our Bertso database and in NUS database. We have selected methods that are suitable to work with segments of different duration as it is the case of Bertso database. On the one hand, we have trained GMM classifiers with the parameters suggested in [12] (ΔF0) and [13] (DFT of F0 distribution). On the other hand, we have also built a GMM classifier based on MFCC parameters. These methods are explained with more detail in the following subsections.

4.2.1. DFT of F0 distribution

As commented before, F0 provides very useful information to discriminate between speech and singing. The histogram of F0 gives information about the range and the distribution of F0 values that for singing voice will be concentrated around the frequencies corresponding to musical notes. In [13] GMMs are used to model the DFT of the F0 distribution and detect the deviation of the instantaneous pitch value from its mean. F0 values in Bertso database range from 75 to 500 Hz. 100 bin histogram is calculated where each segment corresponds to approximately 0.027 octaves. The histogram is normalized to have unit area and then modelled using 8 component GMMs for singing voice and speech.

4.2.2. Delta F0

The dynamics of F0 are another feature to consider in speech and singing voice discrimination. In [12] the ΔF0 distribution of voiced segments is modelled with GMMs to discriminate speech and singing voice. We calculate the ΔF0 using a Savitsky-Golay filter [22] with a window of 50 ms. An histogram of 100 bins is made from -50 to 50 Hz. The distribution of ΔF0 in each voice segment is normalized to have unit area and then modelled with 16 component GMMs.

4.2.3. MFCC GMM

In [12] short-term spectrum features are used motivated by the presence of an additional resonance characteristic of singing speech addressed in [23]. We calculated 13 MFCC coefficients and their Δ with a frame period of 10 ms and a window of 25 ms applying a CMVN file-wise normalization. MFCC frames of speech and singing voice are modelled using 32 component GMMs. Each voice segment is assigned the class that gets the higher sum of log-likelihood for all the frames of the segment. We chose GMMs to model MFCC parameters because of the high dimensionality of the parameters.

4.3. Results of GMM-HMM VAD

The metric used to assess the VAD is the voice detection F-score defined as indicated in equation 7.

\[
F = \frac{2TP}{2TP + FP + FN}
\]

where TP is the duration speech classified as speech, FP is the duration of non-speech classified as speech and FN is the duration of speech classified as non-speech.

Table 2 shows the results for different number of Gaussian components. All of them get good results, over 0.96, and the number of components does not affect the performance significantly. We have selected the VAD with 32 components for the classification experiments.

4.4. Results of speech/singing classification

The metric in each test will be the unweighted F-score, defined as the average of F-score for each of the classes. This metric is suitable when the classes are unbalanced as it is the case of Bertso database. The results of the cross-validation classification experiments are shown in Table 3. The proposed method gets the best results and can compete with spectrum based methods. The methods that use pitch derived parameters as ΔF0 and DFT of F0 distribution get poorer results due to the short duration of the segments to classify. Although in other works they have proved useful, they are not suitable for the characteristics of Bertso database. The experiment in the NUS database shows similar results proving the validity of our method with professional singers and a different style. The GMM method results got worse for NUS database, probably because the audio files used to generate the models contain both speech and singing and the files in NUS database belong to one class. Therefore the normalization process affects both databases differently.

Table 2: Results of the GMM-HMM VAD for different number of Gaussian components

<table>
<thead>
<tr>
<th>Gaussians</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.965 +/- 0.005</td>
</tr>
<tr>
<td>4</td>
<td>0.967 +/- 0.006</td>
</tr>
<tr>
<td>8</td>
<td>0.969 +/- 0.007</td>
</tr>
<tr>
<td>16</td>
<td>0.972 +/- 0.008</td>
</tr>
<tr>
<td>32</td>
<td>0.973 +/- 0.008</td>
</tr>
<tr>
<td>64</td>
<td>0.974 +/- 0.008</td>
</tr>
</tbody>
</table>

Table 3: Results of speech/singing classification

<table>
<thead>
<tr>
<th>Method</th>
<th>Bert. NUS</th>
<th>NUS</th>
<th>Bert. NUS</th>
<th>NUS</th>
<th>Bert. NUS</th>
<th>NUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ΔF0</td>
<td>0.78</td>
<td>0.76</td>
<td>0.83</td>
<td>0.74</td>
<td>0.80</td>
<td>0.74</td>
</tr>
<tr>
<td>DFT-F0</td>
<td>0.73</td>
<td>0.77</td>
<td>0.77</td>
<td>0.76</td>
<td>0.75</td>
<td>0.77</td>
</tr>
<tr>
<td>GMM</td>
<td>0.95</td>
<td>0.75</td>
<td>0.85</td>
<td>0.66</td>
<td>0.89</td>
<td>0.64</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.91</td>
<td>0.89</td>
<td>0.93</td>
<td>0.89</td>
<td>0.92</td>
<td>0.89</td>
</tr>
</tbody>
</table>

5. Conclusions

An efficient singing/speech discrimination system that is suitable for classifying short segments has been developed. It uses only two parameters extracted from pitch to take the decision and classify each segment. As a by-product a flexible algorithm to label musical notes has also been produced. Currently we are testing LSTMs to segment singing and speech taking advantage of their capacity to classify sequences of different lengths.

6. Acknowledgements

This work has been partially supported by UPV/EHU (Ayudas para la Formación de Personal Investigador), the Spanish Ministry of Economy and Competitiveness with FEDER support (MINECO/FEDER, UE) (RESTORE project, TEC2015-67163-C2-1-R) and by the Basque Government under grant KK-2018/00014 (BerbaOla).
7. References


Self-Attention Linguistic-Acoustic Decoder

Santiago Pascual\textsuperscript{1}, Antonio Bonafonte\textsuperscript{1}, Joan Serra\textsuperscript{2}

\textsuperscript{1}Universitat Politècnica de Catalunya, Barcelona, Spain
\textsuperscript{2}Telefónica Research, Barcelona, Spain
\texttt{santi.pascual@upc.edu}

Abstract

The conversion from text to speech relies on the accurate mapping from linguistic to acoustic symbol sequences, for which current practice employs recurrent statistical models like recurrent neural networks. Despite the good performance of such models (in terms of low distortion in the generated speech), their recursive structure tends to make them slow to train and to sample from. In this work, we try to overcome the limitations of recursive structure by using a module based on the transformer decoder network, designed without recurrent connections but emulating them with attention and positioning codes. Our results show that the proposed decoder network is competitive in terms of distortion when compared to a recurrent baseline, whilst being significantly faster in terms of CPU inference time. On average, it increases Mel cepstral distortion between 0.1 and 0.3 dB, but it is over an order of magnitude faster on average. Fast inference is important for the deployment of speech synthesis systems on devices with restricted resources, like mobile phones or embedded systems, where speaking virtual assistants are gaining importance.

Index Terms: self-attention, deep learning, acoustic model, speech synthesis, text-to-speech.

1. Introduction

Speech synthesis makes machines generate speech signals, and text-to-speech (TTS) conditions speech generation on input linguistic contents. Current TTS systems use statistical models like deep neural networks to map linguistic/prosodic features extracted from text to an acoustic representation. This acoustic representation typically comes from a vocoding process of the speech waveforms, and it is decoded into waveforms again at the output of the statistical model \cite{1}.

To build the linguistic to acoustic mapping, deep network TTS models make use of a two-stage structure \cite{2}. The first stage predicts the number of frames (duration) of a phoneme to be synthesized with a duration model, whose inputs are linguistic and prosodic features extracted from text. In the second stage, the acoustic parameters of every frame are estimated by the so-called acoustic model. Here, linguistic input features are added to the phoneme duration predicted in the first stage. Different works use this design, outperforming previously existing statistical parametric speech synthesis systems with different variants in prosodic and linguistic features, as well as perceptual losses of different kinds in the acoustic mapping \cite{3, 4, 5, 6, 7}. Since speech synthesis is a sequence generation problem, recurrent neural networks (RNNs) are a natural fit to this task. They have thus been used as deep architectures that effectively predict either prosodic features \cite{8, 9} or duration and acoustic features \cite{10, 11, 12, 13, 14}. Some of these works also investigate possible performance differences using different RNN cell types, like long short-term memory (LSTM) \cite{15} or gated recurrent unit \cite{16} modules.

In this work, we propose a new acoustic model, based on part of the Transformer network \cite{17}. The original Transformer was designed as a sequence-to-sequence model for machine translation. Typically, in sequence-to-sequence problems, RNNs of some sort were applied to deal with the conversion between the two sequences \cite{18, 19}. The Transformer substitutes these recurrent components by attention models and positioning codes that act like time stamps. In \cite{17}, they specifically introduce the self-attention mechanism, which can relate elements within a single sequence without an ordered processing (like RNNs) by using a compatibility function, and then the order is imposed by the positioning code. The main part we import from that work is the encoder, as we are dealing with a mapping between two sequences that have the same time resolution. We however call this part the decoder in our case, given that we are decoding linguistic contents into their acoustic codes. We empirically find that this Transformer network is as competitive as a recurrent architecture, but with faster inference/training times.

This paper is structured as follows. In section 2, we describe the self-attention linguistic-acoustic decoder (SALAD) we propose. Then, in section 3, we describe the followed experimental setup, specifying the data, the features, and the hyper-parameters chosen for the overall architecture. Finally, results and conclusions are shown and discussed in sections 4 and 5, respectively. The code for the proposed model and the baselines can be found in our public repository \footnote{\url{https://github.com/santi-pdp/musa_tts}}.

2. Self-Attention Linguistic-Acoustic Decoder

To study the introduction of a Transformer network into a TTS system, we employ our previous multiple speaker adaptation (MUSA) framework \cite{14, 20, 21}. This is a two-stage RNN model influenced by the work of Zen and Sak \cite{13}, in the sense that it uses unidirectional LSTMs to build the duration model and the acoustic model without the need of predicting dynamic acoustic features. A key difference between our works and \cite{13} is the capacity to model many speakers and adapt the acoustic mapping among them with different output branches, as well as interpolating new voices out of their common representation. Nonetheless, for the current work, we did not use this multiple speaker capability and focused on just one speaker for the new architecture design on improving the acoustic model.

The design differences between the RNN and the transformer approaches are depicted in figure 1. In the MUSA framework with RNNs, we have a pre-projection fully-connected layer with a ReLU activation that reduces the sparsity of linguistic and prosodic features. This embeds the mixture $x_i$ of different input types into a common representation $h_i$ in the
form of one vector per time step \( t \). Hence, the transformation \( \mathbb{R}^L \rightarrow \mathbb{R}^H \) is applied independently at each time step \( t \) as

\[
h_t = \max(0, Wx_t + b),
\]

where \( W \in \mathbb{R}^{H \times L}, b \in \mathbb{R}^H, x_t \in \mathbb{R}^L, \) and \( h_L \in \mathbb{R}^H \). After this projection, we have the recurrent core formed by an LSTM layer of size \( H \) and an additional LSTM output layer. The MUSA-RNN output is recurrent, as this prompted better results than using dynamic features to smooth cepstral trajectories in time [13].

Based on the Transformer architecture [17], we propose a pseudo-sequential processing network that can leverage distant element interactions within the input linguistic sequence to predict acoustic features. This is similar to what an RNN does, but discarding any recurrent connection. This will allow us to process all input elements in parallel at inference, hence substantially accelerating the acoustic predictions. In our setup, we do not face a sequence-to-sequence problem as stated previously, so we only use a structure like the Transformer encoder which we call a linguistic-acoustic decoder.

The proposed SALAD architecture begins with the same embedding of linguistic and prosodic features, followed by a positioning encoding system. As we have no recurrent structure, and hence no processing order, this positioning encoding system will allow the upper parts of the network to locate their operating point in time, such that the network will know where it is inside the input sequence [17]. This positioning code \( e \in \mathbb{R}^H \) is a combination of harmonic signals of varying frequency:

\[
c_{t, 2i} = \sin \left( \frac{t}{10000} \pi \right)
\]

\[
c_{t, 2i+1} = \cos \left( \frac{t}{10000} \pi \right)
\]

where \( i \) represents each dimension within \( H \). At each time-step \( t \), we have a unique combination of signals that serves as a time stamp, and we can expect this to generalize better to long sequences than having an incremental counter that marks the position relative to the beginning. Each time stamp \( c_t \) is summed to each embedding \( h_t \), and this is input to the decoder core.

The decoder core is built with a stack of \( N \) blocks, depicted within the dashed blue rectangle in figure 1. These blocks are the same as the ones proposed in the decoder of [17], but we only have self-attention modules to the input, so it looks more like the Transformer encoder. The most salient part of this type of block is the multi-head attention (MHA) layer. This applies \( h \) parallel self-attention layers, which can have a more versatile feature extraction than a single attention layer with the possibility of smoothing intra-sequential interactions. After the MHA comes the feed-forward network (FFN), composed of two fully-connected layers. The first layer expands the attended features into a higher dimension \( d_o \), and this gets projected again to the embedding dimensionality \( H \). Finally, the output layer is a fully-connected dimension adapter such that it can convert the hidden dimensions \( H \) to the desired amount of acoustic outputs, which in our case is 43 as discussed in section 3.2. As stated earlier, we may slightly degrade the quality of predictions with this output topology, as recurrence helps in the output layer capturing better the dynamics of acoustic features. Nonetheless, this can suffice our objective of having a highly parallelizable and competitive system.

3. Experimental Setup

3.1. Dataset

For the experiments we use utterances of speakers from the TCSTAR project dataset [22]. This corpora includes sentences and paragraphs taken from transcribed parliamentary speech and transcribed broadcast news. The purpose of these text sources is twofold: enrich the vocabulary and facilitate the selection of the sentences to achieve good prosodic and phonetic coverage. For this work, we choose the same male (M1) and female (F1) speakers as in our previous works. These two speakers have the most amount of data among the available ones. Their amount of data is balanced with approximately the following durations per split for both: 100 minutes for training, 15 minutes for validation, and 15 minutes for test.

3.2. Linguistic and Acoustic Features

The decoder maps linguistic and prosodic features into acoustic ones. This means that we first extract hand-crafted features out of the input textual query. These are extracted in the label format, following our previous work in [20]. We thus have a combination of sparse identifiers in the form of one-hot vectors, binary values, and real values. These include the identity of phonemes within a window of context, part of speech tags, distance from syllables to end of sentence, etc. For more detail we refer to [20] and references therein.

For a textual query of \( N \) words, we will obtain \( M \) label vectors, \( M \geq N \), each with 362 dimensions. In order to inject these into the acoustic decoder, we need an extra step though. As mentioned, the MUSA testbed follows the two-stage structure: (1) duration prediction and (2) acoustic prediction with the amount of frames specified in first stage. Here we are only working with the acoustic mapping, so we enforce the duration with labeled data. For this reason, and similarly to what we did in previous works [14, 21], we replicate the linguistic label vector of each phoneme as many times as dictated by the ground-truth annotated duration, appending two extra dimensions to the 362 existing ones. These two extra dimensions correspond to (1) absolute duration normalized between 0 and 1, given the training data, and (2) relative position of current phoneme inside the absolute duration, also normalized between 0 and 1.

We parameterize the speech with a vocoded representation using Ahocoder [23]. Ahocoder is an harmonic-plus-noise high quality vocoder, which converts each windowed waveform frame into three types of features: (1) mel-frequency cepstral coefficients (MFCCs), (2) log-F0 contour, and (3) voicing frequency (VF). Note that F0 contours have two states: either they follow a continuous envelope for voiced sections of speech, or they are 0, for which the logarithm is undefined. Because of that, Ahocoder encodes this value with \(-10^9\), to avoid numerical undefined values. This result would be a cumbersome output distribution to be predicted by a neural net using a quadratic regression loss. Therefore, to smooth the values out and normalize the log-F0 distribution, we linearly interpolate these contours and create an extra acoustic feature, the unvoiced-voiced flag (UV), which is the binary flag indicating the voiced or unvoiced state of the current frame. We will then have an acoustic vector with 40 MFCCs, 1 log-F0, 1 VF, and 1 UV. This equals a total number of 43 features per frame, where each frame window has a stride of 80 samples over the waveform. Real-numbered linguistic features are Z-normalized by computing statistics on the training data. In the acoustic feature outputs, all of them are normalized to fall within \([0, 1]\).
Figure 1: Transition from RNN/LSTM acoustic model to SALAD. The embedding projections are the same. Positioning encoding introduces sequential information. The decoder block is stacked $N$ times to form the whole structure replacing the recurrent core.

FFN: Feed-forward Network. MHA: Multi-Head Attention.

Table 1: Different layer sizes of the different models. Emb: linear embedding layer, and hidden size $H$ for SALAD models in all layers but FFN ones. HidRNN: Hidden LSTM layer size. $d_f$: Dimension of the feed-forward hidden layer inside the FFN.

<table>
<thead>
<tr>
<th>Model</th>
<th>Emb</th>
<th>HidRNN</th>
<th>$d_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small RNN</td>
<td>128</td>
<td>450</td>
<td>-</td>
</tr>
<tr>
<td>Small SALAD</td>
<td>128</td>
<td>-</td>
<td>1024</td>
</tr>
<tr>
<td>Big RNN</td>
<td>512</td>
<td>1300</td>
<td>-</td>
</tr>
<tr>
<td>Big SALAD</td>
<td>512</td>
<td>-</td>
<td>2048</td>
</tr>
</tbody>
</table>

3.3. Model Details and Training Setup

We have two main structures: the baseline MUSA-RNN and SALAD. The RNN takes the form of an LSTM network for their known advantages of avoiding typical vanilla RNN pitfalls in terms of vanishing memory and bad gradient flows. Each of the two different models has two configurations, small (Small RNN/Small SALAD) and big (Big RNN/Big SALAD). This intends to show the performance difference with regard to speed and distortion between the proposed model and the baseline, but also their variability with respect to their capacity (RNN and SALAD models of the same capacity have an equivalent number of parameters although they have different connexion topologies). Figure 1 depicts both models’ structure, where only the size of their layers (LSTM, embedding, MHA, and FFN) changes with the mentioned magnitude. Table 1 summarizes the different layer sizes for both types of models and magnitudes.

Both models have dropout [24] in certain parts of their structure. The RNN models have it after the hidden LSTM layer, whereas the SALAD model has many dropouts in different parts of its submodules, replicating the ones proposed in the original Transformer encoder [17]. The RNN dropout is 0.5, and SALAD has a dropout of 0.1 in its attention components and 0.5 in FFN and after the positioning codes.

Concerning the training setup, all models are trained with batches of 32 sequences of 120 symbols. The training is in a so-called stateful arrangement, such that we carry the sequential state between batches over time (that is, the memory state in the RNN and the position code index in SALAD). To achieve this, we concatenate all the sequences into a very long one and chop it into 32 long pieces. We then use a non-overlapped sliding window of size 120, so that each batch contains a piece per sequence, continuous with the previous batch. This makes the models learn how to deal with sequences longer than 120 outside of train, learning to use a conditioning state different than zero in training. Both models are trained for a maximum of 300 epochs, but they trigger a break by early-stopping with the validation data. The validation criteria for which they stop is the mel cepstral distortion (MCD; discussed in section 4) with a patience of 20 epochs.

Regarding the optimizers, we use Adam [25] for the RNN models, with the default parameters in PyTorch ($\text{lr} = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$). For SALAD we use a variant of Adam with adaptive learning rate, already proposed in the Transformer work, called Noam [17]. This optimizer is based on Adam with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$ and a learning rate scheduled with

$$\text{lr} = H^{-0.5} \cdot \min(s^{-0.5}, s \cdot w^{-1.5})$$

where we have an increasing learning rate for $w$ warmup training batches, and it decreases afterwards, proportionally to the inverse square root of the step number $s$ (number of batches). We use $w = 4000$ in all experiments. The parameter $H$ is the inner embedding size of SALAD, which is 128 or 512 depending on whether it is the small or big model as noted in table 1. We also tested Adam on the big version of SALAD, but we did not observe any improvement in the results, so we stick to Noam following the original Transformer setup.
Table 2: Male (top) and female (bottom) objective results. A: voiced/unvoiced accuracy.

<table>
<thead>
<tr>
<th>Model</th>
<th>#Params</th>
<th>MCD [dB]</th>
<th>F0 [Hz]</th>
<th>A [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small RNN</td>
<td>1.17 M</td>
<td>5.18</td>
<td>13.64</td>
<td>94.9</td>
</tr>
<tr>
<td>Small SALAD</td>
<td>1.04 M</td>
<td>5.92</td>
<td>16.33</td>
<td>93.8</td>
</tr>
<tr>
<td>Big RNN</td>
<td>9.85 M</td>
<td>5.15</td>
<td>13.58</td>
<td>94.9</td>
</tr>
<tr>
<td>Big SALAD</td>
<td>9.66 M</td>
<td>5.43</td>
<td>14.56</td>
<td>94.5</td>
</tr>
<tr>
<td>Small RNN</td>
<td>1.17 M</td>
<td>4.63</td>
<td>15.11</td>
<td>96.8</td>
</tr>
<tr>
<td>Small SALAD</td>
<td>1.04 M</td>
<td>5.25</td>
<td>20.15</td>
<td>96.4</td>
</tr>
<tr>
<td>Big RNN</td>
<td>9.85 M</td>
<td>4.73</td>
<td>15.44</td>
<td>96.9</td>
</tr>
<tr>
<td>Big SALAD</td>
<td>9.66 M</td>
<td>4.84</td>
<td>19.36</td>
<td>96.6</td>
</tr>
</tbody>
</table>

Table 2 shows the objective results for the systems detailed in section 3.3 over the two mentioned speakers, M and F. For both speakers, RNN models perform better than the SALAD ones in terms of accuracy and error. Even though the smallest gap, occurring with the SALAD biggest model, is 0.3 dB in the case of the male speaker and 0.1 dB in the case of the female speaker, showing the competitive performance of these non-recurrent structures. On the other hand, Figure 3 depicts the inference speed on CPU for the 4 different models synthesizing different utterance lengths. Each dot in the plot indicates a test file synthesis. After we collected the dots, we used the RANSAC [26] algorithm (Scikit-learn implementation) to fit a linear regression robust to outliers. Each model line shows the latency uprise trend with the generated utterance length, and RNN models have a way higher slope than the SALAD models. In fact, SALAD models remain pretty flat even for files of up to 35 s, having a maximum latency in their linear fit of 5.45 s for the biggest SALAD, whereas even small RNN is over 60 s. We have to note that these measurements are taken with PyTorch [27] implementations of LSTM and other layers running over a CPU. If we run them on GPU we notice that both systems can work in real time. It is true that SALAD is still faster even in GPU, however the big gap happens on CPUs, which motivates the use of SALAD when we have more limited resources.

4. Results

In order to assess the distortion introduced by both models, we took three different objective evaluation metrics. First, we have the MCD measured in decibels, which tells us the amount of distortion in the prediction of the spectral envelope. Then we have the root mean squared error (RMSE) of the F0 prediction in Hertz. And finally, as we introduced the binary flag that specifies which frames are voiced or unvoiced, we measure the accuracy (number of correct hits over total outcomes) of this binary classification prediction, where classes are balanced by nature. These metrics follow the same formulations as in our previous works [14, 20, 21].

5. Conclusions

In this work we present a competitive and fast acoustic model replacement for our MUSA-RNN TTS baseline. The proposal, SALAD, is based on the Transformer network, where self-attention modules build a global reasoning within the sequence of linguistic tokens to come up with the acoustic outcomes. Furthermore, positioning codes ensure the ordered processing in substitution of the ordered injection of features that RNN has intrinsic to its topology. With SALAD, we get on average over an order of magnitude of inference acceleration against the RNN baseline on CPU, so this is a potential fit for applying text-to-speech on embedded devices like mobile handsets. Further work could be devoted on pushing the boundaries of this system to alleviate the observed flatter pitch behavior.

6. Acknowledgements

This research was supported by the project TEC2015-69266-P (MINECO/FEDER, UE).
7. References


Japánol: a mobile application to help improving Spanish pronunciation by Japanese native speakers

Cristian Tejedor-García, Valentín Cardeñoso-Payo, David Escudero-Mancebo

Grupo de Investigación ECA-SIMM. Universidad de Valladolid

Abstract

In this document, we describe the mobile application Japánol\(^1\), a learning tool which helps pronunciation training of Spanish as a foreign language (L2) at a segmental level. The tool has been specifically designed to be used by native Japanese people, and implies a branch of a previous CAPT gamified tool TipTopTalk\(^2\). In this case, a predefined cycle of actions related to exposure, discrimination and production is presented to the user, always under the minimal-pairs approach to pronunciation training. It incorporates freely available ASR and TTS and provides feedback to the user by means of short video tutorials, to reinforce learning progression.

Index Terms: computer-assisted pronunciation training, speech recognition, human-computer interaction, computational para-linguistics.

1. Introduction

The way we teach and learn foreign languages is adapting to the technologies. The development of new modes to engage people in learning, as Computer-Assisted Pronunciation Training (CAPT), Computer-Assisted Language Learning (CALL) and Mobile-Assisted Language Learning (MALL), allow improving linguistic skills anytime and anywhere [1]. Also, these systems can give useful information on how learners perform and improve their pronunciation [2]. While there are many software tools that rely on speech technologies for providing to users L2 pronunciation training in the field of Computer Assisted Pronunciation Training (CAPT), Japánol [3], distinctively incorporates a well designed cycle of all the relevant activities related to pronunciation training: exposure, discrimination, production and mixed mode.

This application represents an evolution of previous serious games [4, 5, 6] designed for pronunciation training of L2 by non-native. All of them rely on the minimal pairs methodology [7] and are within the context of research projects related to the development and testing of software tools and games for foreign language learning (TIN2014-59852-R and VA050G18).

Previous versions were based on the free selection by users of exposure, discrimination and production tasks of, mainly, English or Spanish minimal-pairs, in order to get achievements and increase points in leader-boards. With that approach, we were able to assess user’s pronunciation level in a L2 [8]. We have also analyzed how the introduction of corrective feedback [9, 10] increased pronunciation improvement among users after the first stages of use.

\(^1\)This work has been partially supported by the Ministerio de Economía y Empresa (MINECO) and the European Regional Development Fund FEDER under project (TIN2014-59852-R) and by Consejería de Educación de Junta de Castilla y León under project (VA050G18).

2. System’s description

While the freedom of movement on game-oriented tools leads users to maximize their score by repeating those tasks they found easy, the continued use of the tool seemed to generate stagnation. Complementary, learning-oriented tools should focus on users’ difficulties, offering guided and corrective feedback and achieving better effectiveness and efficiency pedagogical results [11]. This motivated this alternate version of the tool, which provides a fixed cycle of well known and balanced learning activities, and which has been applied initially to an experiment which results are presented in this same conference [12].

Figure 1 shows the architecture diagram of components in Japánol. It is a native Android application built from scratch and can be run using low-cost resources available at language laboratories such as computers, tablets, speakers and microphones. It uses Google speech technology, such as Text-To-Speech offline tool and Automatic Speech Recognition web service for Android, that offers a N-best list of probable results for each utterance. Japánol keeps record of chronological events and results of the users with the system in log files. Both audio and log files are stored in a server, through a set of web services, in order to be later analyzed to extract results and conclusions. Lists of Minimal pair words are available in a database accessible to Japánol.

3. Using the tool

Most of current CAPT systems offer isolated pronunciation or discrimination activities as part of the training exercises. Very few combine these different modes as we do in our learning application. In Japánol we follow a learning methodology based on the Theory, Exposure, Discrimination, Pronunciation and Mixed modes.

Figure 1: Components of the CAPT system.
The activities are organized as a sequence of lessons, each devoted to a specific segmental pronunciation difficulty associated to a minimal pair. In each lesson, a brief and clear explanation about the problem and valid pronunciations is provided, in the form of audiovisual material. Then, an exposure mode is entered, in which the user can listen to reference realizations of each valid utterance. After exposure, a discrimination mode is faced in which a sequence of 10 pairs of distinct words (part of a minimal pair) is presented while the user is required to select which one corresponds best with the listened utterance (generated using the TTS). A minimum number of 6 correct pronunciations is to be obtained in order to proceed to the next step. If not, the user is suggested to return to exposure mode again before facing a new discrimination challenge for the same lesson. Once the minimum required number of right answers has been given, or after a number of 5 tries to avoid discouraging the user, a pronunciation mode is entered. In this mode, the user has to say, in sequence, 10 different words selected from the list associated to the minimal pair in the lesson. The recorded speech is submitted to Google Speech ASR and is accepted as valid only when the first item of the N-best list provided by the ASR matches the target word. A minimum number of 6 correct pronunciations is required to pass. If the attempt fails, the user is recommended to return to exposure mode before attempting again.

Finally, a mixed mode activity is required for each lesson. In this mode, a sequence of 10 random discrimination and production tasks are presented and a 60% success is again required to proceed. This mode resembles the added difficulty found to switch from discrimination to production in a normal conversation. A diagram showing the sequence of activities to complete a lesson is shown in Figure 2.

4. Activities in the demonstration

The demonstration will consist on an interactive session showing all different modes in the client application (see 3). People will be able to ask for help during the presentation. At the beginning, all attending people can download the application with a given URL or taking a photo of a QR picture. Once downloaded, the demonstration begins logging into the application before entering the menu of lessons.

5. Acknowledgements

Special thanks to the people of the Language Center of Universidad de Valladolid for their support for the evaluation campaign with Japanese students. We also thank members of the research team for the active support: Enrique Cámara, César González Ferreras and Mario Corrales Astorgano from Universidad de Valladolid, and Antonio Ríos Mestre and María Machuca Ayuso from Universitat Autònoma de Barcelona. Special thanks also to Takuya Kimura, from University of Seisen in Tokyo, who provided Japanese versions of the screens and conducted the experiment with Japanese students at his university.

6. References

Towards the Application of Global Quality-of-Service Metrics in Biometric Systems

J. M. Espín¹, R. Font¹, J. F. Inglés-Romero¹ and C. Vicente-Chicote²

¹Biometric Vox S.L., Spain
²Universidad de Extremadura, Quercus Software Engineering Group, EPCC, Spain

{jm.espin,roberto.font,juanfran.inges}@biometricvox.com, cristinav@unex.es

Abstract

Performance metrics, such as Equal Error Rate or Detection Cost Function, have been widely used to evaluate and compare biometric systems. However, they seem insufficient when dealing with real-world applications. First, these systems tend to include an increasing number of subsystems, e.g. aimed at spoofing detection or information management. As a result, the aggregation of new capabilities (and their interactions) makes the evaluation of the overall performance more complex. Second, performance metrics only offer a partial view of the system quality in which non-functional properties, such as user experience, efficiency or reliability, are generally ignored. In this paper, we introduce RoQME, an Integrated Technical Project funded by the EU H2020 RobMoSys project. RoQME aims at providing software engineers with methods and tools to deal with system-level non-functional properties, enabling the specification of global Quality-of-Service (QoS) metrics. Although the project is in the context of robotics software, the paper presents potential applications of RoQME to enrich the way in which performance is evaluated in biometric systems, focusing specifically on Automatic Speaker Verification (ASV) systems as a first step.

Index Terms: speaker recognition, automatic speaker verification, non-functional properties

1. Introduction

Assessing the performance of biometric systems usually involves: (1) selecting or creating an evaluation corpus, (2) defining a testing protocol, (3) applying the defined corpus and protocol to the system under consideration, and (4) computing a number of performance metrics. These metrics are generally aimed at capturing the ability of the system to produce discriminative and well-calibrated scores. The development of common corpora, protocols and metrics has been one of the major initiatives and well-calibrated scores. The development of non-functional properties, enabling the specification of global Quality-of-Service (QoS) metrics. Although the project is in the context of robotics software, the paper presents potential applications of RoQME to enrich the way in which performance is evaluated in biometric systems, focusing specifically on Automatic Speaker Verification (ASV) systems as a first step.

Properties such as usability or security are generally referred to as non-functional properties. They can be considered as particular quality aspects of a system and represent an essential part of any software solution. Unfortunately, non-functional properties often get missed when designing most software systems. In this vein, the RoQME project [6], which is currently in progress, aims at enabling the modeling (at design time) and estimation (at runtime) of non-functional properties, particularly in the field of robotics. This paper introduces potential applications of the RoQME project to biometric systems, in particular, the paper focuses on ASV systems as a first attempt to address the problem. To the best of our knowledge, the evaluation of biometric systems in terms of global Quality-of-Service (QoS) metrics, defined on non-functional properties, has not been addressed before.

2. The RoQME project

The RoQME Integrated Technical Project (ITP), funded by EU H2020 RobMoSys project [7], aims at providing software engineers with a model-driven tool-chain allowing them to: (1) model system-level non-functional properties in terms of contextual information; and (2) generate ready-to-use code to evaluate QoS metrics, defined on the properties previously specified. Regarding the former, RoQME will provide engineers with a high-level language to model context variables (e.g., battery level) and, from them, relevant context patterns (e.g., “the battery level drops more than 1% per minute”). The detection of a context pattern will be considered an observation in a Bayesian Network, which is a probabilistic model used in RoQME to express the dynamics of a non-functional property (e.g., power consumption).

Once the model containing the relevant properties, contexts and observations is specified, RoQME automatically generates the code that will be responsible for estimating the degree of fulfillment of each non-functional property in terms of QoS metrics. Basically, the implementation will consist of: (1) a context model that receives raw contextual data and produces context events (e.g., changes in the battery level); (2) an event processor that searches for the event patterns specified in the RoQME
model and, when found, produces observations (e.g., battery is draining too fast); and, finally (3) a probabilistic reasoner that computes (based on Bayesian inference) a numeric estimation for each metric (e.g. a value of 0.89 can be understood as the probability of being optimal in terms of power consumption).

Although the RoQME project is focused on the robotics domain, the modeling tools being developed are designed to be extensible and application domain agnostic. Further details about the RoQME project can be found in [8, 9].

3. Applications to biometric systems

This section describes some potential applications of RoQME to biometric systems, in particular, to ASV systems. As stated before, although the RoQME project is focused on the robotics domain, we believe that it explores an issue of great relevance for many other software systems.

3.1. Benchmarking

Using performance metrics, such as Equal Error Rate or Detection Cost Function, to quantify the goodness of the results (either scores or binary decisions) is an essential instrument to evaluate and contrast different algorithms and technologies. However, when it comes to the integral quality of a real-world system, these metrics fail to capture some important aspects, including:

• The interaction of all the different subsystems and their combined effect on False Acceptance and False Rejection ratios.

• The effect of usability and accessibility considerations on Failure to Acquire, Failure to Enroll, and False Rejection ratios. For example, if the instructions on the screen are not clear enough, the user might be unable to provide a valid voice sample. In this case, the performance of the system would be affected before any sample reaches the verification process.

• The impact of certain system configurations and decisions at design/development time, such as the thresholds of the different subsystems or how many authentication attempts are granted to verify an user. Regarding the latter, it is obvious that a system with no limits on the number of failed authentication attempts would be less secure than an identical system with appropriate limits.

We propose to integrate non-functional properties (i.e., security, usability, reliability, etc.) into the system quality evaluation. In this sense, RoQME would allow software engineers to benchmark complex biometric systems, in terms of their overall performance, given a number of QoS metrics. Basically, engineers would start specifying the required non-functional properties by using the high-level modeling language provided by RoQME. After that, the resulting models will automatically generate the code of a software component. At runtime, this component will continuously update the QoS metrics associated with the specified properties. Finally, this information would be considered together with other performance metrics (e.g. Equal Error Rate) to compare different systems, configurations, etc.

3.2. Dynamic assessment

Monitoring QoS metrics over time would allow engineers to detect any anomaly or deviation from the expected behavior and foresee future problems. In this sense, RoQME could be useful for the early identification of malfunctioning or undesired behaviors. At runtime, RoQME can provide system administrators with real-time QoS indicators about the degree of fulfillment in usability, resource utilization or stability, to name just a few examples. This information can then be used by the engineers to improve the system or adjust its configuration.

3.3. Self-adaptation

Speaker verification systems could use the QoS metrics provided by RoQME to automatically adjust its own configuration (or software) in order to optimize its performance under changing circumstances. For instance, this approach could be applied to estimate the voiceprint quality. This would allow the system to dynamically change its operation, e.g., to ask the user to provide additional voice samples if his/her voiceprint quality falls. Furthermore, RoQME QoS metrics could play an important role in an unsupervised strategy for adapting voiceprints, aimed at gradually improving their quality as the system is used. This approach would make the system more robust, e.g., against aging effects.

4. Acknowledgements

RoQME has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N. 732410, in the form of financial support to third parties of the RobMoSys project.

5. References


Incorporation of a Module for Automatic Prediction of Oral Productions Quality in a Learning Video Game

David Escudero-Mancebo, Valentin Cardeñoso-Payo

1Department of Computer Science, University of Valladolid, Spain
descuder@infor.uva.es, valen@infor.uva.es

Abstract

This document presents the research project TIN2017-88858-C2-1-R of the Spanish Government. Antecedents and goals of the project are presented. Current status, recent achievements and collaborations after the first year development are described in the paper.

Index Terms: Computer Assisted Pronunciation Training, Prosody, Down syndrome speech, Learning video games.

1. Antecedents

This project continues the research line opened in 2014 with the project La piedra mágica1 supported by Resercaixa and continued in 2016 with the project Pradia2 supported by BBVA Humanidades Digitales. The first of these projects served to develop a video game for Down syndrome adolescents to train oral communication, in particular prosody and pragmatics [1]. In the second project, the video game routines had been enriched and a clinical evaluation of user’s performance was tested with formal evaluation.

2. Goals of the project

The video games developed in previous projects require the permanent assistance of a personal trainers (a teacher, a therapist, a relative… ) to control and monitor the progress of the game users. The goal of the current project is programming a module to increase the software capabilities so that users can play in an autonomous way supervised by an intelligent tutor. This intelligent system will be responsible to decide on the user to repeat or continue with the training activities giving feedback for him/her to be more competent in oral communication and prosodic skills.

Three stages have been defined: 1) corpus collection of audio of DS people playing with the video game; the DS audios will be rated by professional voice therapists; correcting advices of the therapists will be also monitored. 2) computational models of quality of DS turns will be trained from the audio; a knowledge data base will be compiled with the therapist corrective orders. 3) An expert system is using the compiled information to decide about the user activities in real time.

3. Current status

The following activities have been performed according to the programed schedule:

• The software has been adapted to record in real time annotations from trainers about the quality of the video game users’ oral turns.

• An app for mobile devices has been programmed to collect trainers annotations in a transparent way for players.

• An evaluation template has been devised including a set of criteria to be evaluated by the trainers. This template has been designed in collaboration with voice therapists from the centers Colegio Tórtila de Educación Especial of Fundación Personas Valladolid and ASDOVA Down Syndrome Association Valladolid.

• An inter-rater consistency evaluation process is being run for assessing the evaluation template and guaranty the consistency of parallel evaluation to be performed during the sort term recording activity.

4. Achievements

Main achievement in this first year of the project development is the publication [2] which reports on results derived from data obtained in previous projects. The video game was also presented in [3]. The PhD thesis of Mario Corrales, framed in this project, was presented in XXXIV Congreso SEPLN Sevilla Septiembre 2018 (the SEPLN society granted the student to attend the conference). In the current conference we also present advanced on the rating of quality of DS voice from data [4].

5. Collaborations

Apart from the already reported collaborations with Fundación Personas Valladolid and ASDOVA, we have open several national and international promising collaborations.

On one hand, we are collaborating with the research group of Pastora Martínez from the Department of Psychology of UNED Madrid. She is co-author of several relevant publications in the field of Down syndrome and prosody [5, 6]. She has provided as with her paired corpus of typical-DS speech, obtained while applying the PEPS-C test [7].

On the other hand, we submitted an Erasmus+ project (strategic partnership) in collaboration with different research groups and education centers in Portugal, Ireland, Italy, France and Bosnia. The project was finally rejected but the evaluation gives encouraging feedback for future resubmissions.

6. Acknowledgments

Special thanks to Yolanda Martín de San Pablo from Fundación Personas, Alfonso Rodríguez from ASDOVA and Jesús Gómez from CEIP Uruela for the collaboration during the evaluation template definition and during the inter-rater consistency tests.

Thank’s to all the team for their active support: Valle Flores, Carlos Vivaracho, César González, Lourdes Aguilar and Mario Corrales.
7. References


Silent Speech: 
Restoring the Power of Speech to People whose Larynx has been Removed

Jose A. Gonzalez1, Phil D. Green2, Damian Murphy3, Amelia Gully3, and James M. Gilbert4

1University of Malaga, Spain
2University of Sheffield, U.K.
3University of York, U.K.
4University of Hull, U.K.

j.gonzalez@uma.es

Abstract

Every year, some 17,500 people in Europe and North America lose the power of speech after undergoing a laryngectomy, normally as a treatment for throat cancer. Several research groups have recently demonstrated that it is possible to restore speech to these people by using machine learning to learn the transformation from articulator movement to sound. In our project articulator movement is captured by a technique developed by our collaborators at Hull University called Permanent Magnet Articulography (PMA), which senses the changes of magnetic field caused by movements of small magnets attached to the lips and tongue. This solution, however, requires synchronous PMA-and-audio recordings for learning the transformation and, hence, it cannot be applied to people who have already lost their voice. Here we propose to investigate a variant of this technique in which the PMA data are used to drive an articulatory synthesiser, which generates speech acoustics by simulating the airflow through a computational model of the vocal tract. The project goals, participants, current status, and achievements of the project are discussed below.

Index Terms: speech restoration, silent speech interfaces, permanent magnet articulography, articulatory synthesis, magnetic resonance imaging

1. Introduction

A total laryngectomy is a clinical procedure in which the voice box is surgically removed most commonly as a treatment for throat cancer. This procedure not only leaves the subject muted, but it is also known to cause social isolation, feelings of loss of identity and can lead to clinical depression [1, 2, 3]. Current available methods for speaking after a laryngectomy include the electro-larynx, a hand-held device which produces an unnatural, electronic voice; oesophageal speech, which is difficult to master, and the voice prosthesis, which is considered to be the current gold standard, but has a short life time (4 to 8 weeks) due to candida growth, thus requiring regular hospital visits for valve replacement [4, 5, 6]. Other available methods such as the Alternative and Augmentative Communication (AAC) devices [7], where the user types words and the device synthesises them, are also limited by their slow manual input and, therefore, are not suitable for any other than short conversations.

As an alternative to existing speech restoration methods, we are investigating a new way to restore speech to those who are unable to speak [8, 9, 10, 11, 12]. The idea is to transform measurements of the lips and tongue movements obtained using magnetic sensing into audible speech using a speaker-dependent transformation, implemented by machine learning techniques (typically deep neural networks [13]). For capturing articulator movement we use Permanent Magnet Articulography (PMA) [14, 15, 16, 17], a technology developed by our collaborators at the University of Hull in which small magnets are attached to the lips and tongue and the magnetic field generated when the articulators move is captured by sensors close to the mouth (see Fig. 1 for a picture of the PMA system). The parameters of the transformation for converting articulator movement into speech are currently estimated from simultaneous recordings of audio and PMA signals acquired before the person loses her/his voice. Some audio samples produced by the proposed restoration method can be found at https://www.jandresgonzalez.com/is2017. As can be seen, the samples are mostly intelligible and the speaker identity is clearly preserved.

A limitation of the above approach for speech restoration is that simultaneous speech-and-sensor recordings are required for estimating the mapping between articulator movement and its acoustics. Thus, this makes this method unsuitable for persons who have already lost their voice. The aim is of this project is, thus, to investigate a novel approach that would make simultaneous recordings unnecessary. The idea is to predict, in real time, the position of the speech articulators from the PMA signals. This is a non-trivial problem as the magnetic field arriving at the sensors is a composite of the fields generated by all the magnets attached to the articulators. From the estimated vocal tract shapes speech can finally be synthesised by simulating airflow propagation through the vocal tract using well-known techniques.
established articulatory synthesis methods [18].

In the next sections, the detailed objectives of the project, the participants, and its current status are described in detail.

2. Project objectives

As previously mentioned, the goal of this project is to investigate and develop a new method for speech restoration based on the PMA capturing technique and machine learning, but without the need of parallel speech and sensor recordings for training the machine learning techniques. We attempt to do this by, instead, learning an alternative transformation which will map the articulatory data captured by the PMA device into a physical model of the vocal tract (e.g. 1D or 2D representation of the vocal tract). Then, we will be able to generate audible speech from the estimated vocal tract shapes by using well-known articulatory synthesis methods.

The detailed objectives of this project are:

- To train a direct transformation from PMA data to vocal tract shapes used by the articulatory synthesiser.
- To personalise the synthesiser so that the speech generated sounds like the users original voice.

With regard to the latter point, we expect to use MRI images of the subject’s vocal tract to personalise the synthesiser. In this way, the acoustics generated by the synthesiser will resemble the user’s original voice.

3. Partners

There are four partners involved in this project:

- University of Sheffield: Jose A. Gonzalez (principal investigator; now at the University of Malaga), Phil D. Green and Roger K. Moore.
- University of York: Damian Murphy, Helena Daffern and Amelia Gully.
- University of Hull: James M. Gilbert and Lam A. Cheah.
- University of Leeds: Andy Bulpitt and Duane Carey.

4. Current status

Firstly, we have been able to record a database consisting of parallel recordings of MRI, PMA and speech data for 4 subjects. For this pilot study, we decided to record simple material, mainly consonant-vowel (CV) and vowel-consonant-vowel syllables. At the same time, we have been working on improving the quality of speech generated by our articulatory synthesiser. In this regard, we described in [19] a dynamic 3D digital waveguide mesh (DWM) vocal tract model capable of movement to produce diphthongs. In [20], we further investigated on estimating a physical model of the vocal tract (a 2D model in this case) from the speech waveform, rather than magnetic resonance imaging data. As an advantage, this method provides a clear relationship between the model and the size and shape of the vocal tract, offering considerable flexibility in terms of speech characteristics such as age and gender. Finally, we have been also working on optimizing the appearance and usability of the PMA system.

5. Acknowledgements

This work was supported by the White Rose university consortium through a Collaboration Fund grant. We gratefully ac-

knowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

6. References


RESTORE Project: REpair, STOrage and REhability of speech

Inma Hernández¹, Eva Navas¹, Jose Antonio Municio², Javier Gómez²

¹UPV/EHU
²HUC-Biocruces

inma.hernandez@ehu.eus, eva.navas@ehu.eus
joseantonio.municiomartin@osakidetza.eus, jgomezsuarez@aeorl.net

Abstract

RESTORE is a project aimed to improve the quality of communication for people with difficulties producing speech, providing them with tools and alternative communication services. At the same time, progress will be made at the research of techniques for restoration and rehabilitation of disordered speech. The ultimate goal of the project is to offer new possibilities in the rehabilitation and reintegration into society of patients with speech pathologies, especially those laryngectomised, by designing new intervention strategies aimed to favour their communication with the environment and ultimately increase their quality of life.

Index Terms: alaryngeal voice, oesophageal speech, speaking aids, voice rehabilitation, statistical parametric speech synthesis, voice bank

1. Introduction

According to the Spanish Statistical Institute (Instituto Nacional de Estadística) (data from 2012), there are more than 410 000 people with ‘disability to produce spoken messages’ in Spain. This kind of disability, if severe, produces social isolation since the communication with the environment is seriously affected so as to hamper personal relationships and generate problems of integration at the workplace. The origin and displaying forms of the oral disabilities is varied. They might be caused by traumatic events such as stroke or surgery like Total Laryngectomy (TL) or also by degenerative diseases (such as ELA or Parkinson) which deteriorate the functioning of the motor speech system. This project is concerned with two specific groups of affected persons: on the one hand, people with a laryngectomy and on the other hand, people with dysarthria suffering from poor articulation of phonemes due to neurological injury of the motor speech system.

In the Rehabilitation Service of the Cruces University Hospital (the main public Hospital of the area of Bilbao), almost 6000 speech therapy sessions are performed annually. Among them, around 50 patients are treated after a total laryngectomy. Even though nowadays laryngeal cancer is treated with chemotherapy and radiotherapy to preserve the laryngeal function, in some cases total laryngectomy must be performed [1]. In 2014 laryngeal cancer had an incidence in Spain of 3182 per 100 000 inhabitants with a mortality of 1.3% and a 5-year prevalence of 112 per 1000 inhabitants [2]. After total laryngectomy patients must attend a number of speech therapy sessions to acquire a new voice, generally called alaryngeal voice. In the Cruces University Hospital all patients start the rehabilitation treatment 2 or 3 weeks after hospital discharge.

People affected by ELA are frequent users of Text to Speech Conversion systems, usually integrated in AAC systems (Alternative and Augmentative Communication). These AAC systems are designed to help the user to quickly build messages to be spoken out loud (with a keyboard or more sophisticated input devices). Commercial AAC systems have usually a limited choice of the synthetic voice: they are usually high quality voices sounding like a very healthy young person. Nevertheless, statistics show a reality in which a great majority of the people affected are elderly people, whose real voice would not match with the prosthetic one. Similarly, there is a lack of children’s voices, and according to the data in Spain there are 32 700 children between 6 and 15 years affected by this disability.

RESTORE is a project aimed to improve the quality of communication for people with difficulties producing speech, providing them with tools and alternative communication services. At the same time, progress will be made at the research of techniques for restoration and rehabilitation of disordered speech. The following goals were proposed for the project:

1. Implementation of a donors voice bank and creation of the service of personalised synthetic speech for people with oral disabilities in general and people who have been laryngectomised, in particular at Cruces University Hospital.
2. Design and development of a Serious Game as a multimedia tool to support the speech rehabilitation process for people with voice disorders.
3. Improvement of the intelligibility of alaryngeal voices (voices from people who have been laryngectomised) and dysarthric voices.

The Project coordinates the research work of two groups, one from the field of Signal Processing and Engineering and the other from the fields of Otorhinolaryngology and Speech Therapy. The Voice Banking service proposed is a groundbreaking service for the country, since there are only a few experiences with the same aim in USA (VocalID¹, ModelTalker²) and Europe (SpeakUnique³) [3]. This kind of service offers personalised synthetic speech that can be used in existing AAC devices. In our proposal the communication is achieved with a personalised voice that might be similar to the patients own voice in cases of progressive diseases or laryngectomies. Besides, the design and development of a voice training tool is pursued in order to incentive and facilitate the speech rehabilitation process, a real hard and complex task depending on the specific pathology.

In this paper we describe the main tasks of the project and briefly sketch the main results up to date.

¹www.vocalid.co
²www.modeltalker.com
³www.speakunique.org
2. Tools to support the speech therapist

Total Laryngectomy surgery completely removes the larynx of the patient while separating the airway from the mouth, nose and oesophagus. Consequently, patients who undergo a TL can not produce speech sounds in a conventional manner because their vocal cords have been removed. The rehabilitation process of a patient starts immediately upon confirmation of the surgery. Through a pre-surgical interview an orientation framework is offered both to the patient and his or her family where they will receive information about:

- The anatomical and physiological changes that result from surgery
- The way to communicate during the period immediately following surgery
- The speech therapy sessions that will follow surgery

The main objective of the rehabilitation after TL is to return to the patient the possibility of oral communication for reintegration into their social, work and personal life. After surgical intervention, additionally to medical treatment and the importance of tracheostomy protection and care of the tracheal cannula the patient will be informed about the therapy process to acquire alaryngeal voice. There are three possibilities for voice rehabilitation:

- Oesophageal speech
- Tracheoesophageal speech
- Use of an electrolarynx

Oesophageal speech (ES) is preferred by medical doctors because it does not require a voice prosthesis, but it is also most effortful and difficult to acquire. Tracheoesophageal speech is the most successful method and also produces the most understandable speech, but requires a voice prosthesis placed during total laryngectomy or later in a secondary puncture. Finally, the electrolarynx is an external vibrating handheld device which is placed to the neck or the face. The vibrating sound is modulated by the movements of the articulators to produce understandable speech. It produces a robotic voice and it is sometimes used also as a backup secondary method.

In Cruces University Hospital laryngectomised patients start rehabilitation after 2 or 3 weeks after hospital discharge with the aim of learning to produce oesophageal speech. The patient attends around 50 rehabilitation sessions during a period of 4 months. If the final speaking method is tracheoesophageal, the average learning period is only 5 days.

In RESTORE we have developed an interactive video aimed at helping the patient during and after this rehabilitation period. This video considers the main difficulties faced by the laryngectomised patient and proposes exercises and advices to overcome them. Using a comic style representation of a food market, the main character represents the patient itself, going through the different market stalls, in each of which he or she will practice a new rehabilitation exercise. Video recordings of real sessions with a speech therapist are also included, as well as short interviews with laryngectomised persons that have succeeded in the rehabilitation process and share their own feelings and experiences.

Clinical evaluation of the developed tool is currently taking place with laryngectomised patients.

3. Personalisation of the synthetic voice

One of the goals of RESTORE project is to improve the already existing ZureTTS voice bank web portal making it more flexible and allowing to provide a personalised TTS service to people with oral disabilities.

3.1. Voice bank and web portal

In the previous version of the voice bank web portal, each voice donor had to record 100 sentences to get his or her personalised voice. This is not a problem for healthy people, but many patients are not able to record such a long corpus. Therefore the original 100 sentences corpus has been divided into three corpora of 33, 33 and 34 sentences. Each donor can choose how many corpora to record and the personalised voice will be produced with the available speech material. Besides, two new languages have been incorporated to the portal: Gaelic and the Navarro-Lapurdian dialect of Basque.

3.2. Recording protocol for pre-laryngectomised people

To be able to generate a personalised voice for laryngectomises, the ideal situation is to have recordings of the patient made prior to the surgery. If this is not the case, recordings made by close family members can be used to produce a personalised synthetic voice for the patient, as voices are usually similar among family members of the same gender [4] [5].

The first step to get these recordings is to introduce the recording procedure in the hospital protocol. This protocol has established the following criteria to select patients that can take part in the project:

1. Patients older than 18 years old with a TL programmed.
2. Close family members with voices similar to the patient.
3. Any patient older than 18 years old without any speech pathology who comes to the otolaryngology service of the Cruces University Hospital for any reason.

Once these criteria have been fixed, the protocol has been sent to the Basque Ethics Committee for approval.

3.3. Personalisation for dysarthric voices

One person with ELA make use of the ZureTTS portal to obtain a synthetic voice. However, the voice was already affected by the disease and resulting synthetic voice also showed the same problems of the original voice. The main issue was the rhythm, which was very slow, with very long vowels and very frequent long pauses, thus confirming other studies [6][7]. Also, some consonants were poorly realised. The slow prosody provoked the malfunctioning of the automatic alignment algorithm thus contributing to the low quality of the synthetic voice. Nevertheless, even when a new specific alignment method adapted to the multiple pauses was applied, the synthetic voice was not of the desired quality, mainly because of the long phones. To overcome it, prosody transplantation from a healthy voice was performed with good results. This voice was offered to the user, instead of the one automatically provided by the system. We also tried to palliate the pronunciation problems applying the adaptation techniques using only vowels described in [8][9], but the new synthetic voice, although with an improved pronunciation lost the personality of the speaker. We are also experimenting with model surgery on some phones [10], but the improvements are subtle.

aholab.ehu.eus/zureTTS
3.4. Synthetic voices catalogue

If the person with oral disabilities is not able to record the corpus and there is no family member with a similar voice, he or she is still able to get a customised synthetic voice, by selecting the one of his or her preference among all the donated voices.

To make the selection of a personalised voice easier, a catalogue with the available voices has been included in the WEB portal. A subjective evaluation where listeners qualified all the synthetic voices according to several attributes was developed. These attributes were: white - hoarse voice, sweet - dominant voice, warm - high-pitched voice, clean - nasal voice and monotonous - expressive voice. A bidimensional representation of the voices using the two most discriminative dimensions (sweet-dominant and white-hoarse) has been integrated in the portal, as shown in Figure 1.

The voices can be easily modified in tone, rhythm, intensity and vocal tract length to get a synthetic voice that pleases the user. The customised voices can be obtained from the web portal in standard format for Android OS, iOS and Windows, so they can be directly used by Augmentative and Alternative Communication Devices.

4. Voice conversion

In the production of oesophageal speech the pharyngo-oesophageal segment is used as a substitutive vibrating element for the vocal folds. Due to the nature of the intervention, the air used to create the vibration of the oesophagus can not come from the lungs and the trachea as happens during normal speech production. Instead, the air is swallowed from the mouth and introduced in the oesophagus, being then expelled in a controlled way while producing the vibration. These huge differences in the production mechanisms lead to a diminution of naturalness and intelligibility [11][12][13]. As a consequence, the communication with others is hindered. Moreover, these less intelligible voices are an added problem for the automatic speech recognition algorithms that are becoming ubiquitous in the human computer interaction technologies. One of the goals of this project is the development of techniques and algorithms aimed at modifying oesophageal speech in such a way as to improve the performance of a state of the art ASR system with these modified signals as input. To achieve this goal we decided to experiment with voice conversion (VC) algorithms. In this section we summarize the efforts done within the project in this direction.

4.1. ASLABI Database

To our knowledge, there are not publicly available databases for oesophageal Spanish speech. There are studies published concerning the quality and characteristics of ES, some or them for Spanish, but they use their own recording data, mostly developed for the purpose of the specific research [14][15][16][17]. Additionally, most VC algorithms make use of parallel databases. The availability of recorded voices for the 100 phonetically balanced sentences of ZureTTS for healthy speech made us decide the recording of the same set of sentences for oesophageal speakers. To make the recordings, we contacted the local Association Asociación de Laringectomizados de Bizkaia who showed an enormous interest in the project and collaborated with enthusiasm. A total of 32 persons went to make the recordings to the acoustically isolated room in our Faculty. The database also includes one Basque session.

4.2. Improving the intelligibility of the oesophageal voice

We have tried several strategies to improve the intelligibility of the oesophageal speech. First, we tried a classical GMM based voice conversion, using parallel data. This was followed by DNN based approaches, using LSTM and more recently also including a WaveNet vocoder.

There are several specific problems that must be faced when processing and converting oesophageal signals:

- Oesophageal signals lack the regular periodicity (intonation) typical of laryngeal signals. Although they have a certain periodicity at certain segments, the fundamental frequency is very low (around 80Hz) and very irregular. Usual F0 calculation algorithms generate many errors and do not result in a realistic measure of the local periodic segments. Thus a specific F0 detection algorithm has been used.

- The rhythm in general, the duration of syllables and the duration of the phones inside them, vary significantly in relation to healthy speech. Additionally, noises and pauses are inserted in between words and syllables even inside the syllable. Therefore, the alignment of healthy and oesophageal parallel sentences becomes a tricky task.

- Many phones (mainly corresponding to consonants) in the sounds stream are not present in the signal or they are realised in a completely different way. This fact also complicates the parallel alignment task and generates many recognition errors.

- In general, a fundamental frequency curve must be estimated for the converted spectrum (except for the case of using WaveNet). Simple conversion of the source F0 values as usually done in the VC field is not feasible, which opens a wide range of research possibilities.

To evaluate the intelligibility of the resulting converted signals we have used a Kaldi-based ASR system [18] trained with material described in [19]. This approach was selected because it allows us to control the exact processing operations followed
during the recognition (such as the use of transformations like fMLLR) as well as basic aspects of the recognition process such as the lexicon and the language model. This turned out to be very important with our reduced set of 100 phonetically rich sentences, containing many very unlikely words, proper names etc. The procedure and results of the different ASR tests are described in [20][21].

5. Discussion and future work

This project represents an effort to promote modern technological advances in the area of speech processing among the group of people with oral disabilities. In particular we have put special emphasis to introduce the benefits of the advances in speech synthesis in the rehabilitation process of the laryngectomised people.

In relation to the intelligibility of oesophageal speech, we plan to improve the techniques to evaluate not only human intelligibility but also the effort employed by the listener (‘listening effort’).

At the time of writing this paper, only two laryngectomised persons have been recorded for the voice bank previous to TL. It must be taken into account that the elapsed time between the communication of the need to do the TL surgery and the surgery itself is very short (usually of a few days). So there is not the necessary time for the medical team to explain the patient the future benefits of making the recordings. Additionally, these patients have their voice already very harsh and speak with difficulties (it is usually the reason why they have contacted the unit). This is why we consider the catalogue of synthetic voices, possibly including voices of the patient’s relatives as a very good alternative. We hope that this voice bank pilot experiment will continue growing and expanding the service to other hospitals after the end of the present project.

6. Acknowledgements

This project has been founded by the Spanish Ministry of Economy and Competitiveness with FEDER support (RESTORE project, TEC2015-67163-C2-1-R and TEC2015-67163-C2-2-R).

7. References


Corpus for Cyberbullying Prevention

Asunción Moreno, Antonio Bonafonte, Igor Jauk, Laia Tarrés, Victor Pereira
Innovation and Technology Center of the Universitat Politècnica de Catalunya

Abstract

Cyberbullying is the use of digital media to harass a person or group of people, through personal attacks, disclosure of confidential or false information, among other means. That is to say, it is considered cyberbullying, or cyber-aggression to everything that is done through electronic communication devices with the intended purpose of harming or attacking a person or a group.

In this paper we present a starting project to prevent cyberbullying between kids and teenagers. The idea is to create a prevention system. A system which is installed in the mobile of a kid and, if a harassment is detected, some advice is given to the child. In case of serious or repeated behavior the parents are alerted.

The focus of this paper is to describe the characteristics of the database to be used to train the system.

Index Terms: bullying, cyberbullying, databases, children

1. Introduction

The fight against bullying and cyberbullying among children and adolescents is becoming a priority. A study by Save the Children published by EL PAÍS\(^1\) reveals that 1 out of 10 children during ESO has suffered harassment or cyberbullying in Spain.

The most frequent types of harassment are direct or indirect insults, spreading rumors, damage to property, physical damage, exclusion and threats.

According to the Ministry of Health, INE and OMS, in 2015, the number of suicides of children under 15 years was 12 and in the range of 15 to 29 years was 247. On the other hand, the number of suicide attempts of children under 15 years old was 273, of whom the majority, 234, were girls.

d-LAB is a program of Mobile World Capital Barcelona whose objective is to carry out Calls for Proposals that serve to promote responses to social problems through the implementation of collaborative pilot projects between private companies and public entities.

Earlier this year, d-LAB launched a call entitled "Fighting cyberbullying through mobile technologies"\(^2\). The company SafeToNet (STN) was the winner through the "Safeguarding children online" project.

STN proposed a service to get ahead of a child harassing another person via mobile and alert their parents. The service includes warning about sexually improper photographs, and identification of bullying.

The Pilot Project began in July 2018. The Project partners are: d-LAB, SafeToNet (London, UK), Orange and the Innovation and Technology Center of the Universitat Politècnica de Catalunya (CIT-UPC).

The Project consists of several phases:

- Generation of a database in Spanish and Catalan for the training of cyberbullying prevention systems.
- Adaptation of the System currently available in English to Catalan and Spanish.
- Test in several Catalan schools with children between 12 and 14 years old chosen in a way that provides a diverse socio-economic environment to cover cultural varieties.
- Validation and improvement based on the system test on Orange employees and their family members.

This paper deals with the generation of a database in Spanish and Catalan for the training of cyberbullying prevention systems.

2. Database specifications

Two databases have been created, one in Catalan and another in Spanish spoken in Spain. Each database consists of 140,000 posts, of which 100,000 are labeled by two people and 40,000 by a single person.

The posts are labeled according to their content in 7 categories: aggression, anxiety, depression, distress, sexuality, use of substances, and violence. For each category, 5 levels of concern have been established. 1: post nothing worrying, 5: post extremely worrying.

The annotation has been made through a platform available in STN in which there is an automatic system to present the posts to be scored and an efficient annotation system based on keyboard or mouse interchangeably.

\(^1\)https://elpais.com/elpais/2016/02/18/media/1455822566_899475.html
\(^2\)https://d-lab.tech/challenge-3/
3. Recruiting of labelers and technical support

17 people were selected and hired for the labelling task. Most of the annotators are students. They have different backgrounds. Following recommendations from SafeToNet, most of the labelers are psychology students. There are 4 male and 13 female annotators, all of them between 20 and 30 years old. Table 1 shows the background summary of the annotators:

<table>
<thead>
<tr>
<th>CODE</th>
<th>BACKGROUND</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Philosophy student</td>
</tr>
<tr>
<td>F2</td>
<td>Teacher degree</td>
</tr>
<tr>
<td>F3</td>
<td>Psychology student</td>
</tr>
<tr>
<td>F4</td>
<td>Criminology student</td>
</tr>
<tr>
<td>F5</td>
<td>Biomedicine student</td>
</tr>
<tr>
<td>F6</td>
<td>Psychology student</td>
</tr>
<tr>
<td>F7</td>
<td>Social Communication student</td>
</tr>
<tr>
<td>F8</td>
<td>Psychologist degree</td>
</tr>
<tr>
<td>F9</td>
<td>Biology and Neuroscience student</td>
</tr>
<tr>
<td>F10</td>
<td>Statistics and economy student</td>
</tr>
<tr>
<td>F11</td>
<td>Children’s education</td>
</tr>
<tr>
<td>F12</td>
<td>Psychology student</td>
</tr>
<tr>
<td>F13</td>
<td>Engineering student</td>
</tr>
<tr>
<td>M1</td>
<td>Engineering student</td>
</tr>
<tr>
<td>M2</td>
<td>Psychology student</td>
</tr>
<tr>
<td>M3</td>
<td>Physical activity and sports degree</td>
</tr>
<tr>
<td>M4</td>
<td>Engineering student</td>
</tr>
</tbody>
</table>

Table 1. Code and Background of the annotators

Annotators were contracted on a part time basis (20 hours/week). They work from home. They have some flexibility to manage their dedication to the project during each week. However, their daily dedication to the project can never be higher than 5 hours. This dedication was set to avoid tiredness and lack of concentration during the annotation procedure.

A training week (16th–20th July) was organized at UPC premises. A person from SafeToNet was in charge of the training during the first three days. Annotators were instructed, one category at a time, with several examples chosen from real posts. The last two days, students had a simulation of real work using the annotation platform reaching the labeling of 400 posts each day. Those posts were carefully chosen to show simultaneously several categories and several concern levels. Another re-training week was necessary at the mid part of the project to consensus inter-annotator agreement.

4. Compilation and selection of posts

Posts were selected from several sources such as twitter, teenagers’ chats, blogs, forums, medical consultation web sites, etc. Data was manually or automatically downloaded, cleaned, formatted and selected. Posts were chosen to have a minimum number of characters (without counting –not discarding- @names and internet addresses) of 50 and a total maximum of 280. Spanish as spoken in Latin America posts were discarded when possible. As expected, Catalan data was harder to collect. The number of information in internet in Spanish is huge compared against Catalan websites, blogs and consulters. In addition, Catalan speakers are bilingual, and it is very common to find Spanish posts in Catalan sites. Spanish and Catalan can be naturally mixed even in the same post.

5. Quality Assessment

To calculate the consistency between the annotators we used three indices: Cohen’s kappa index, Accuracy, and Cronbach’s alpha index.

5.1. Cohen’s kappa index

The Cohen’s kappa index [1] between two annotators labeling the same data measures the consistency between the annotations and compares them with the case of the annotation being random. To calculate the consistency between two annotators, the common messages labeled by both of them are searched and the results of the annotation are compared for each characteristic. The Cohen’s kappa index is calculated as:

\[ K = \frac{p_o - p_e}{1 - p_e} \]

Where \( p_o \) is the relative agreement between raters (accuracy), and \( p_e \) is the probability of chance agreement. Cohen’s kappa index was calculated on binarized categories: (category A: level of concern 1 or 2; category B: level of concern 3, 4, or 5)

5.2. Cronbach’s alpha coefficient

The Cronbach alpha coefficient has been calculated per each category \( y \) as

\[ \alpha_y = \frac{K}{K-1} \left[ 1 - \frac{\sum_{i=1}^{2} \sigma^2_{yi}}{\sigma^2_y} \right] \]

Where: \( K \): number of items (K=2: 2 annotators) \( \sigma^2_{yi} \): variance of each item (i.e vector of length nposts of category y of tagger i) \( \sigma^2_y \): variance of the total (i.e. vector: \( t=y_1+y_2 \) of length nposts)

6. Results

The table shows a lower inter agreement in anxiety and distress. This is a consequence of no direct translation of distress within Spanish and Catalan. The annotators had difficulties to distinguish both categories.

7. Discussion

The data collection will be finished on October 15th so that the first prototype will be ready one month later. The results of the complete project will be presented in Mobile World Congress 2019.

8. Acknowledgements

We want to thank d-LAB and STN for the trust placed in us to carry out the project.

9. References

EMPATHIC, Expressive, Advanced Virtual Coach to Improve Independent Healthy-Life-Years of the Elderly


1Speech Interactive Group, Universidad del País Vasco UPV/EHU, Spain
2Intelligent Voice, United Kingdom
3Intelligent System Group, Universidad del País Vasco UPV/EHU, Spain
4Osatek, Spain
5Acapela Group, Belgium
6Universita degli Studi della Campania Luigi Vanvitelli, Italy
7Management Center Innsbruck, Austria
8Institut Mines Télécom, France
9Tunstall Healthcare, Sweden
10Oslo Universitetssykehus HF, Norway
11E-Seniors, France
12Technion, Israel

manes.torres@ehu.eus, coordinator@empathic-project.eu

Abstract

The EMPATHIC Research & Innovation project researches, innovates, explores and validates new paradigms and platforms, laying the foundation for future generations of Personalised Virtual Coaches to assist elderly people living independently at and around their home.

The project uses remote non-intrusive technologies to extract physiological markers of emotional states in real-time for online adaptive responses of the coach, and advances holistic modelling of behavioural, computational, physical and social aspects of a personalised expressive virtual coach. It develops causal models of coach-user interactional exchanges that engage elders in emotionally believable interactions keeping off loneliness, sustaining health status, enhancing quality of life and simplifying access to future telecare services.

EMPATHIC proposes multidisciplinary research and development, involving:

• Geriatrician, Neuroscientist, Psychiatric, health and social work specialists, knowledgeable in age related conditions and the aims of a coaching program in maintaining independence, functional capacity, and monitoring physical, cognitive, mental and social well-being to implement the individual coaching goals
• Psychologist, Neuroscientist and Computer Science experts for detection and identification of the emotional status of the user
• Engineers and Computer Scientists in speech and language technologies, biometrics, image analysis, and machine learning. They will develop tools to detect emotional cues, model users’ emotional status, translate coaching plans into actions, user-adapt spoken dialogue and personalise talking agents
• Telecare services, a senior association and a hospital interested in testing and validating EMPATHIC
• Companies interested in providing and developing technology for the project and commercialising the products and derived services

Through measurable end-user validation, to be performed in 3 different countries (Spain, Norway and France) with 3 distinct languages and cultures (plus English for R&D), the proposed methods and solutions will ensure usefulness, reliability, flexibility and robustness

Index Terms: speech recognition, natural language understanding, spoken dialog systems, human-computer interaction, natural language generation, text to speech conversion, emotional features from speech and language, emotional voice

1. Objectives

➢ OBJ1: Design a virtual coach, to engage the healthy-senior user and reach pre-set benefits, measured through project-defined metrics, to enhance well-being through awareness of personal physical status, by improving diet and nutritional habits, by developing more physical exercise and by social activity

172
 ➢ **OBJ2:** Involve end-users and to reach a degree of fit to their personalised needs and requirements, derived by the coach, which will enhance their well-being

 ➢ **OBJ3:** Supply the coach with incorporate non-intrusive, privacy-preserving, empathic, expressive interaction technologies

 ➢ **OBJ4:** Validate the coach efficiency and effectiveness across 3 distinct European societies (Norway, Spain, and France), with 200 to 250 subjects – who will be involved from the start

 ➢ **OBJ5:** Evaluate/validate the effectiveness of EMPATHIC designs against relevant user’s personalised acceptance and affordance criteria (such as the ability to adapt to users’ underlying mood) assessed through the Key Performance Indicators (KPI) listed in Section 1.1.2

 ➢ **OBJ6:** Drive the developed methodology and tools to industry acceptance and open-source access identifying appropriate evaluation criteria to improve the “specification-capture-design-implementation” software engineering process of implementing socially-centred ICT products.

### Scientific Goals (Sc) and Research Actions:

1. Provide automatic personalised advice guidance (through the coach) having a direct impact in empowering elder users into a wide of advanced ICT keeping improving their quality of life and level their independent independency living status of the people as the age. EMPATHIC researches a) the identification and assessment of main cues related to physical, cognitive, mental and social well-being b) defining personalised, psychologically motivated and acceptable coach plans and strategies c) translating professional coach behaviour into actions of the Intelligent EMPATHIC-VC.

2. Identifying non-intrusive technologies to detect the individual’s emotional and health status. of the person through non-intrusive technologies. EMPATHIC uses emotional information from eyes, face, speech and language to deliver a hypothesis of the user emotional status to assist decisions of the EMPATHIC-VC. In this framework, the research focus on the detection of sudden shifts in the user emotional status or emotional changes during a certain period of time: a) extraction of emotional features b) data-driven approaches for multimodal modelling combining emotional cues provided by each source c) identification of significant changes in individual behaviour.

3. Implement health-coach goals and actions through an intelligent computational system, intelligent coach and spoken dialogue system adapted to users’ intentions, emotions and context. EMPATHIC researches a) Data-driven modelling of user and tasks; b) Machine learning for understanding the user; c) Learning policies and questionnaires to deal with coach goals; d) Statistical approaches for dialogue management driven by both user and Coach goals; e) Online learning for adaptation.

4. Provide the virtual coach with a natural, empathic, personalised and expressive communication model in a supportive manner to allow emotional bonds that result in engaged and effective relationships. EMPATHIC researches and develop, through early and continuous interaction with the end-users, the technologies to consider individual user profile, including cultural facts and interaction history, the current emotional status of the user and the coach strategies at each decision of the dialogue manager, at each text generated by the Natural Language Generator, at each inflexion of the TTS and at each movement of the personalised visual agent.

### Technological Goals (Tg) and Actions:

1. Develop a simulated virtual coach and acquire an initial corpus of dialogues. A set of annotated dialogues will be designed and obtained through a Wizard-of-Oz (WoZ) technology to fulfil the initial end-users and data requirements of Scientific Goals #2, #3 and #4.

2. Integrate and provide a proof-of-concept of the technology running on different devices

3. Validation through Field trials. EMPATHIC will test representative realistic use cases for different user profiles in three different countries

### 2. An example

**Virtual Coach:** So, Pablo, ¿Have you ever eaten consistently 2 or 3 pieces of fruit?

**User:** When my wife was healthier, she used to take care of buying the fruit. Thus it was easier for me to eat it.

**Virtual Coach:** And in any other time of your life?

**User:** Long time ago, when I lived alone, I used to take care of buying the fruit myself, and I ate it more frequently.

**Virtual Coach:** What does this information suggest to you about your objective of eating 2 or 3 pieces of fruit a day?

**User:** Well… That it’s something that basically depends on me.

**Virtual Coach:** So, can you see anything you could do to get closer to your objective?

**User:** Ummm… I should start thinking how I am going to organise when I am going to buy the fruit.

![Image](image_url)
Advances on the Transcription of Historical Manuscripts based on Multimodality, Interactivity and Crowdsourcing

Emilio Granell, Carlos-D. Martínez-Hinarejos, Verónica Romero

Pattern Recognition and Human Language Technology Research Center, Universitat Politècnica de València, Camí de Vera s/n, 46022, València, Spain
{egranell,cmartine,vromero}@dsic.upv.es

Abstract

The transcription of digitalised documents is useful to ease the digital access to their contents. Natural language technologies, such as Automatic Speech Recognition (ASR) for speech audio signals and Handwritten Text Recognition (HTR) for text images, have become common tools for assisting transcribers, by providing a draft transcription from the digital document that they may amend. This draft is useful when it presents an error rate low enough to make the amending process more comfortable than a complete transcription from scratch.

The work described in this thesis is focused on the improvement of the transcription offered by an HTR system from three scenarios: multimodality, interactivity and crowdsourcing.

The image transcription can be obtained by dictating their textual contents to an ASR system. Besides, when both sources of information (image and speech) are available, a multimodal combination is possible, and this can be used to provide assistive systems with additional sources of information. Moreover, speech dictation can be used in a multimodal crowdsourcing platform, where collaborators may provide their speech by using mobile devices.

Different solutions for each scenario were tested on two Spanish historical manuscripts, obtaining statistically significant improvements.

Index Terms: handwritten text recognition, automatic speech recognition, multimodality, combination, interactivity, crowdsourcing

1. Introduction and Motivation

Transcription of digitised historical documents is an interesting task for libraries in order to provide efficient information access to the contents of these documents. The transcription process is done by experts on ancient and historical handwriting called paleographers.

In the latest years, the use of off-line Handwritten Text Recognition (off-line HTR) systems [1] has allowed to speed up the manual transcription process. HTR systems are composed of modules and employ models similar to those of classical speech recognition systems. However, state-of-the-art off-line HTR systems [2] are far from being perfect, and human supervision is required to really produce a transcription of standard quality. The initial result of automatic recognition may be incorporated to the assistive transcription system, as well as to incorporate the user feedback (on-line HTR).

In addition to using off-line HTR systems from text line images, other modalities of natural language recognition can be used to help paleographers on the transcription process, such as Automatic Speech Recognition (ASR) [3] from the dictation of the contents, or on-line HTR [4] from touchscreen pen strokes. In this context, a multimodal interactive assistive scenario [5], where the assistive system and the paleographer cooperate to generate the perfect transcription, would reduce the time and the human effort required for obtaining the final result.

The use of multimodal collaborative transcription applications (crowdsourcing) [6], where collaborators can employ speech dictation of text lines as a transcription source from their mobile devices, allows for a wider range of population where volunteers can be recruited, producing a powerful tool for massive transcription at a relatively low cost, since the supervision effort of paleographers may be dramatically reduced.

In this thesis [7], the reduction of the required human effort for obtaining the actual transcription of digitalised historical manuscripts is studied in the following scenarios:

• Multimodality: An initial draft transcription of a handwritten text line image can be obtained by using an off-line HTR system. An alternative for obtaining this draft transcription is to dictate the contents of the text line image to an ASR system. Furthermore, when both sources (image and speech) are available, a multimodal combination is possible, and an iterative process can be used in order to refine the draft transcription. Multimodal combination can be used in interactive transcription systems for combining different sources of information at the system input (such as off-line HTR and ASR), as well as to incorporate the user feedback (on-line HTR).

At the same time, the multimodal and iterative combination process can be used to improve the initial off-line HTR draft transcription by using the ASR contribution of different speakers in a collaborative scenario.

• Interactivity: The use of assistive technologies in the transcription process reduces the time and human effort required for obtaining the actual transcription. The assistive transcription system proposes a hypothesis, usually derived from a recognition process of the handwritten text line image. Then, the paleographer reads it and produces a feedback signal (first error correction, dictation, etc.), and the system uses it to provide an alternative hypothesis, starting a new cycle. This process is repeated until a perfect transcription is obtained. Multimodality can be incorporated to the assistive transcription system, in order to improve the human-computer interaction and to provide the system with additional sources of information.

1Publicly available in the UPV institutional repository: http://hdl.handle.net/10251/86137
Crowdsourcing: Open distributed collaboration to obtain initial transcriptions is another option for improving the draft transcription to be amended by the paleographer. However, current transcription crowdsourcing platforms are mainly limited to the use of non-mobile devices, since the use of keyboards in mobile devices is not friendly enough for most users. An alternative, is the use of speech dictation of handwritten text lines as a transcription source in a crowdsourcing platform where collaborators may provide their speech by using their own mobile device. Multimodal combination allows the improvement of the initial handwritten text recognition hypothesis by using the contribution of speech recognition from several speakers, providing as a final result a better draft transcription to be amended by a paleographer with less effort. In this framework, since collaborators are usually a scarce resource, their acquisition effort should be optimised with respect to the quality of the draft transcriptions.

The rest of this paper is structured as follows: Section 2 offers the main scientific and technological goals; Section 3 summarises the contents of this thesis; Section 4 contains the main conclusions; Section 5 draws the current work derived from this thesis and the future work lines; Finally, Section 6 presents the achievements, and the scientific contributions.

2. Scientific and Technological Goals
The main scientific and technological goals of this thesis are the following:

- To study the unimodal and multimodal combination techniques, in order to propose a new multimodal combination technique for improving the transcription of digitised historical manuscripts by using the speech dictation of their contents.
- To study the use of multimodal combination techniques in a computer assisted system to improve the computer-human interaction and to accelerate the interactive transcription process.
- To develop a multimodal crowdsourcing platform based on the studied multimodal combination techniques to ease and widespread the transcription of digitised historical manuscripts.

3. Thesis Overview
The thesis document [7] is structured in five parts to facilitate the reading experience. It starts with a first introductory part, followed by a part for each one of the three studied scenarios, and it finishes with a part which presents the general conclusions and future work lines. This section presents an overview of the contents of the three central parts (multimodality, interactivity, and crowdsourcing).

3.1. Multimodality
The integration of knowledge given by off-line HTR and ASR processes presents two limitations: both signals are asynchronous and each modality uses different basic linguistic units (usually, characters for off-line HTR and phonemes for ASR). An initial approach for solving this limitation was proposed in previous works [8, 9], where the output of the recognition process of one modality, in form of word-graph lattice, is used to modify the general language model in order to make more likely the decoded sentences; this modified language model is employed in the decoding for the other modality. This procedure can be used iteratively. This approach presents a few drawbacks: there is not a single hypothesis given that each modality provides its own, and it is not known beforehand which one is more accurate, and the initial modality must be chosen arbitrarily.

Chapter 3 of the thesis (Combining Handwriting and Speech) presents a new proposal based on the use of Confusion Networks for obtaining a single hypothesis from the combination of the hypotheses obtained from an off-line HTR and an ASR recognisers for decoding a text line image and the dictation of its contents. In the next chapter (Chapter 4), our multimodal proposal is tested and compared with other combination methods.

The experiments were performed on two different Spanish historical manuscripts, Cristo Salvador, which is a single writer book from the 19th century provided by Biblioteca Valenciana Digital, and Rodrigo [10], that corresponds to the digitisation of the book Historia de España del arzobispo Don Rodrigo, which was written in old Castilian (Spanish) in 1545. Both corpora are publicly available for research purposes on the website of the Pattern Recognition and Human Language Technology (PRHLT) research center ². Acoustics models were trained by using the Spanish phonetic corpus Albayzin [11].

The transcriptions quality is assessed using the Word Error Rate (WER) value, which allows us to obtain a good estimation for the paleographer post-edition effort, and the lattices quality by the oracle WER, which represents the WER of the best hypotheses contained in the word lattices (more details about corpora and evaluation metrics can be found in Chapter 2 of the thesis).

Table 1: Summary of the multimodal experimental results.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Cristo Salvador</th>
<th>Rodrigo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WER</td>
<td>Oracle WER</td>
</tr>
<tr>
<td>Off-line HTR</td>
<td>32.9%</td>
<td>27.5%</td>
</tr>
<tr>
<td>ASR</td>
<td>43.3%</td>
<td>27.4%</td>
</tr>
<tr>
<td>Multimodal</td>
<td>29.3%</td>
<td>13.4%</td>
</tr>
</tbody>
</table>

Table 1 summarises the results of the multimodal experiments. As it can be observed, the behaviour is similar for both corpora. The use of the ASR does not improve the WER of the draft offered by the off-line HTR system, although the word-graphs generated offer similar values of oracle WER for both modalities. However, combining both modalities by using our proposal, not only the WER is improved, but the oracle WER value of the multimodal word-graph lattices is substantially reduced. Given that the oracle WER value is related to the quality of the alternatives offered by our interactive and assistive system, an outstanding effect on interactive transcription can be expected.

3.2. Interactivity
The result of combining the knowledge given by off-line HTR and ASR processes may make the paleographer task easier, since they are able to correct on an improved draft transcription. However, given that paleographer revision is required to

²https://prhlt.upv.es/
produce a transcription of standard quality, an interactive assistive scenario, where the automatic system and the paleographer cooperate to generate the perfect transcription, would provide an additional reduction of the human effort and time required for obtaining the final result.

Chapter 5 of the thesis (Assistive Transcription) presents a multimodal interactive transcription system where the paleographer feedback is provided by means of touchscreen pen strokes, traditional keyboard, and mouse operations. The combination of the different sources of information is based on the use of Confusion Networks derived from the decoding output of three recognition systems: two HTR systems (off-line and on-line), and an ASR system. Off-line HTR and ASR are used to derive (by themselves or by combining their recognition results) the initial hypothesis, and on-line HTR is used to provide feedback. In the next chapter (Chapter 6 of the thesis), our multimodal and interactive proposals are tested.

In this case, the interactive performance is given by Word Stroke Ratio (WSR), the definition of which makes it comparable with the WER. The relative difference between them gives us the effort reduction (EFR), which is an estimation of the transcription effort reduction that can be achieved by using the interactive system (see Chapter 2 of the thesis for more details).

Table 2: Summary of the multimodal interactivity experimental results.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Cristo Salvador</th>
<th>Rodrigo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WSR</td>
<td>EFR</td>
</tr>
<tr>
<td>Off-line HTR</td>
<td>30.2%</td>
<td>8.2%</td>
</tr>
<tr>
<td>ASR</td>
<td>35.1%</td>
<td>−6.7%</td>
</tr>
<tr>
<td>Multimodal</td>
<td>14.1%</td>
<td>57.1%</td>
</tr>
</tbody>
</table>

Table 2 summarises the results of the assistive and interactive experiments. As it can be observed, the estimated interactive human effort (WSR) required for obtaining the perfect transcription from the off-line HTR decoding represents about 8% of relative effort reduction (EFR) over the off-line HTR WER for both corpora (see Table 1). However, in the case of ASR no effort reduction can be considered. Regarding multimodality, as expected, the use of the proposed multimodal approach allows the interactive system to achieve more than 30% of relative effort reduction over the off-line HTR WER for both corpora.

Table 3: Summary of the multimodal feedback and interactivity experimental results for Cristo Salvador.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Deletions</th>
<th>WSR</th>
<th>TS</th>
<th>KBD</th>
<th>Global</th>
<th>EFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off-line HTR</td>
<td>5.5%</td>
<td>26.0%</td>
<td>6.7%</td>
<td>32.7%</td>
<td>0.6%</td>
<td></td>
</tr>
<tr>
<td>ASR</td>
<td>5.1%</td>
<td>31.6%</td>
<td>3.5%</td>
<td>35.1%</td>
<td>−6.7%</td>
<td></td>
</tr>
<tr>
<td>Multimodal</td>
<td>1.9%</td>
<td>10.7%</td>
<td>1.3%</td>
<td>12.0%</td>
<td>63.5%</td>
<td></td>
</tr>
</tbody>
</table>

In Table 3 a summary of the multimodal feedback (i.e. on-line handwriting feedback) and interactivity experimental results for Cristo Salvador are presented. In this case, the WSR is calculated under the assumptions that the deletion of words have no cost, and that the cost of keyboard-correcting an erroneous on-line feedback word is similar to another on-line HTR interaction. Therefore, the WSR correspond with the percentage of words written with the on-line HTR feedback (TS) and the percentage of words corrected by means of the keyboard (KBD). As it can be observed, the multimodal combination of the on-line feedback with the input hypotheses reduces significantly the amount of words that are required to be corrected by using the keyboard, and most of the paleographer effort is concentrated in the more ergonomic touchscreen feedback.

3.3. Crowdsourcing

As an alternative to the keyboard, volunteers could employ voice as input for transcription. Nearly all mobile devices provide this modality, which widens the range of population and situations where collaboration can be performed. The main drawback is that the audio transcription, usually obtained by ASR systems, presents an ambiguity not present in typed input. Even the state-of-the-art techniques, although more accurate than a few years ago, produce a considerable amount of errors in the recognition process, which makes it necessary to obtain a balance between the amount of collaborations and the quality they provide.

In any case, the need for final supervision by a paleographer enables the possibility that, although not perfect, voice inputs combined with off-line HTR provide an initial draft transcription more accurate than that given only by off-line HTR. This fact was confirmed with the statistically significantly improvements obtained in the experiments performed for the previous parts of this thesis, multimodal, and interactive transcription. Thus, the employment of speech collaborations will allow us to significantly reduce the final transcription effort.

Chapter 7 of the thesis (Collective Collaboration) explores how a crowdsourcing framework that allows for text line dictations acquisition could decrease the transcription effort. The framework is based on the use of multimodal recognition, both employing and combining off-line HTR and ASR results, to improve the final transcription that is going to be offered to the paleographer. The multimodal recognition approach is based on language model interpolation and Confusion Network combination techniques. The crowdsourcing platform was implemented by using a client-server architecture. The client is a mobile application that allows speech acquisition and the server part performs the recognition and combination operations. In the next chapter (Chapter 8), our multimodal crowdsourcing proposal is tested in a supervised and in an unsupervised mode for the Rodrigo [10] corpus.

In the supervised experiments, the collaborators were randomly sorted 11 times giving 11 different order lists. Figure 1 presents the results of the speaker ordering in the supervised crowdsourcing experiments. Best, worst and the median of 11 different random orders.

Figure 1: Results of the speaker ordering in the supervised crowdsourcing experiments. Best, worst and the median of 11 different random orders.
shows the evolution, from the initial off-line HTR baseline until the process of the speech of the last collaborator, for the lists that obtained the worst, the median and the best final results. As it can be observed, the worst and the best final results do not represent any statistically significant differences. These results show that, in the best case, only two speakers are needed to obtain significant improvements. Meanwhile, in the worst case at least four speakers are needed.

![Graph showing Word Error Rate (WER) for HTR baseline and ASR outputs for the whole unsupervised collaborations. The horizontal lines represent the corresponding average ASR WER.](image)

**Figure 2:** Baseline values and the evolution of the system and ASR outputs for the whole unsupervised collaborations. The horizontal lines represent the corresponding average ASR WER.

From the unsupervised experiments, Figure 2 draws the baseline values for both modalities and the evolution of the system and ASR outputs. As it can be observed, the language model interpolation permits to reduce the error level in the next speech decoding process [8], and the combination with the speech decoding results allows the system output to converge to a better hypothesis with less errors to correct [16]. Besides, the ASR performance is considerably improved, reducing the average WER baseline value. Finally, after processing the speech of the last collaborator, the system outputs presented 25.3% of WER that represents 35.6% of relative statistically significant improvement over the off-line HTR baseline, and an estimated time reduction for the paleographer revision of about 5 minutes per page [10].

4. Main Conclusions

Regarding multimodality, the benefits of multimodal combination of the results obtained from off-line HTR with additional sources of information for the transcription of historical manuscripts have been confirmed.

With respect to interactivity, multimodality was applied on an interactive tool for transcribing historical handwritten documents. On the one hand, the multimodal hypotheses combination allows to reduce the human time and workload required for transcribing historical books, due to the increased recognition accuracy and the better quality of the alternatives contained in the multimodal lattice. On the other hand, the use of multimodal combination allows to improve the human-computer interaction (by using on-line touch-screen handwritten pen strokes), given that the multimodal combination allows to correct errors on the interactive system hypothesis by using the information provided by the on-line handwritten text introduced by the user.

Finally, the proposed multimodal crowdsourcing framework is based on the iterative refinement of the language model and hypotheses combination. This framework uses a client / server architecture in order to allow collaborators to decide when and where to collaborate. The mobile application used for speech acquisition is publicly available [13].

The experiments showed that, in this framework, the number of collaborators is more important than the order in which their speech is processed. Through this experimentation, it has been shown that the use of speech is a good additional source of information for improving the transcription of historical manuscripts, and that this modality allows people to collaborate in this task using their own mobile device.

5. Current and Future Work

Currently, we are testing the performance of our assistive and interactive system with more robust modelling methods based on deep learning.

Regarding multimodality, we propose for future studies the use of whole sentences instead of lines of the handwritten text document because it might make multimodality more natural from the point of view of the paleographer or speaker who has to dictate the contents of the handwritten text images to the ASR system.

In the case of interactive transcription, we have already tested the use of speech not only as an additional source of information of the handwritten text line image to transcribe in the interactive and assistive system, but as an additional modality for human-computer interaction [14]. Furthermore, our future works aim also at taking advantage of the real samples that are produced while the system is used for adapting the feedback natural language recognisers to the user.

Finally, the proposed multimodal crowdsourcing framework and the multimodal interactive transcription system were integrated [15], and in the near future, we are planning to test it with other datasets.

6. Scientific Contributions

The main contributions of this thesis can be summarised in: the evaluation on how to combine the decoding output of different natural language recognition systems, the integration of the combination of different signals in a computer assisted transcription system, and the development of a multimodal crowdsourcing platform for the transcription of historical manuscripts.

The scientific impact of this thesis was supported by eight publications at the time of the dissertation presentation. Concretely, the multimodality part was supported by two articles presented in two international conferences (ICDAR 2015 [16], and CAIP 2015 [17]), the interactivity part by two publications, one in an international conference and the other in a book chapter (DAS 2016 [18], Handwriting, Nova 2017 [19]), and the crowdsourcing part by four publications, two in international conferences, one in a book chapter, and one in a JCR international journal (DocEng 2016 [20], IberSPEECH 2016 [21, 13], IEEE/ACM TASLP [22]).

Moreover, in the time of writing this paper, an additional publication on an international conference is supporting the interactivity part (DAS 2018 [14]), and another in a JCR international journal the crowdsourcing part (COIN [15]).

7. Acknowledgments.

8. References


Bottleneck and Embedding Representation of Speech for DNN-based Language and Speaker Recognition

Alicia Lozano-Diez, Joaquin Gonzalez-Rodriguez, Javier Gonzalez-Dominguez

Audias-UAM, Universidad Autonoma de Madrid, Madrid, Spain
alicia.lozano@uam.es

Abstract

In this manuscript, we summarize the findings presented in Alicia Lozano Diez’s Ph.D. Thesis, defended on the 22nd of June, 2018 in Universidad Autonoma de Madrid (Spain). In particular, this Ph.D. Thesis explores different approaches to the tasks of language and speaker recognition, focusing on systems where deep neural networks (DNNs) become part of traditional pipelines, replacing some stages or the whole system itself. First, we present a DNN as classifier for the task of language recognition. Second, we analyze the use of DNNs for feature extraction at frame-level, the so-called bottleneck features, for both language and speaker recognition. Finally, utterance-level representation of the speech segments learned by the DNN (known as embedding) is described and presented for the task of language recognition. All these approaches provide alternatives to classical language and speaker recognition systems based on i-vectors (Total Variability modeling) over acoustic features (MFCCs, for instance). Moreover, they usually yield better results in terms of performance.

1. Introduction

Lately, automatic speech recognition (ASR) has experienced a breathtaking progress, partially thanks to the introduction of deep neural networks (DNNs) into their approaches. This has spread across related areas such as language identification (LID) and speaker recognition (SID), where DNNs have noticeably improved their performance.

In this manuscript we present a summary of the main findings of the Ph.D. Thesis defended by Alicia Lozano Diez, where we focused on different approaches for LID and SID based on DNNs, replacing some stages or the whole system. The complete dissertation can be found in [1].

First, end-to-end language recognition systems based on DNNs are analyzed, where the network is used as classifier directly. We focus on two architectures: convolutional DNNs (CDNNs) and long short-term memory (LSTM) recurrent neural networks (RNNs), which are less demanding in terms of computational resources due to the reduced amount of free parameters in comparison with other DNNs. Thus, they provide an alternative to classical i-vectors, achieving comparable results, especially when dealing with short utterances.

Second, we explore one of the most prominent applications of DNNs in speech processing: as feature extractors. Here, DNNs are used to obtain a frame-by-frame representation of the speech signal, the so-called bottleneck feature (BNF) vector, which is learned directly by the network and is then used instead of traditional acoustic features as input in LID and SID systems based on i-vectors. This approach revolutionized these two fields, since they highly outperformed state-of-the-art systems (i-vector based on acoustic features). Our analysis focuses on how different configurations of the DNN used as BNF extractor, which is trained for ASR, influences performance of resulting features for LID and SID.

Finally, we propose a novel approach for LID, in which the DNN is used to extract a fixed-length utterance-level representation of speech segments known as embedding, comparable to i-vector, and overcoming the disadvantage of variable length sequence of BNFs. This embedding-based approach has recently shown promising results for SID, and our proposed system was able to outperform a strong state-of-the-art reference i-vector system on the last challenging 2015 and 2017 NIST LREs. We explore different architectures and data augmentation techniques to improve results of our system, obtaining comparable or better results than the well-established i-vectors.

2. DNN as a Classifier for LID

We call end-to-end DNN-based systems to those that perform the target task from the input, without any other backend. For LID, they usually take some input features and are trained to classify each input frame into one of the target languages, outputting the probability vector of a frame belonging to each language. Here, we use CDNNs and LSTMs, which have less parameters than other DNNs. Besides, LSTMs have shown to be a good model for time-depending sequences [2]. We present experiments with both architectures on a balanced subset of 8 languages from LRE 2009, on the 3 s task. We compare our systems to an i-vector baseline, with 1024-dimensional UBM, 400-dimensional i-vectors and cosine scoring.

2.1. Convolutional DNN for Language Recognition

CDNNs usually consist of convolution and subsampling layers [3]. The former aims to perform feature extraction and each of its units is connected to a local subset of units in the previous layer. Groups of these units share their parameters and form a feature map that extracts the same features from different locations in the input. The subsampling layer selects the maximum activation of each region on the input (max-pooling). The general scheme of our CDNN-LID system is depicted in Figure 1.

We use as input 56-dimensional MFCC-SDC [4] feature vectors and segments of 3 s. All the architectures have 3 hidden layers and we vary the number of filters (feature maps) in each one, keeping fixed their shape and max-pooling regions. The output layer is a fully-connected layer with softmax activation, which outputs the probability that a test segment belongs to a certain language. The network is trained with stochastic gradient descent to minimize the negative log-likelihood. We conducted experiments to evaluate the influence of differ-
ent amounts of data used to train, balanced per language. The configurations used are summarized in Table 1.

### 2.1.1. Experiments and Results

Results of this section are summarized in Table 2. Although our standalone CDNN-based systems are outperformed by the i-vector, their size is smaller and are trained with less data. We see that systems benefit from larger training datasets (compare systems 2, 3 and 4) and bigger models (compare ConvNet 3 and 5, or 4 and 6). Moreover, improvements are obtained when fusing the best CDNN system with the i-vector baseline, meaning complementary information extracted from the same features.

### 2.2. LSTM RNN for Language Identification

LSTMs are able to store information from previous inputs during long time periods [5, 6, 7], which makes them more suitable to model sequential data. They replace hidden units in a classical DNN with memory blocks [8], which have input, output and forget gates: the input and output gates control respectively the flow of input activations into the memory cell and the output flow of cell activations into the rest of the network; the forget gate allows the flow of information from the memory block to the cell, adaptively resetting the cell’s memory.

![Figure 1: Representation of a CDNN architecture for LID.](image1)

Table 1: Configuration of the CDNN models for LID.

<table>
<thead>
<tr>
<th>ID</th>
<th># Filters/Layer</th>
<th>Development Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvNet 1</td>
<td>[20, 30, 50]</td>
<td>~178h Train ~31h Validation</td>
</tr>
<tr>
<td>ConvNet 2</td>
<td>[5, 15, 20]</td>
<td>~178h Train ~31h Validation</td>
</tr>
<tr>
<td>ConvNet 3</td>
<td>[5, 15, 20]</td>
<td>~356h Train ~63h Validation</td>
</tr>
<tr>
<td>ConvNet 4</td>
<td>[5, 15, 20]</td>
<td>~534h Train ~63h Validation</td>
</tr>
<tr>
<td>ConvNet 5</td>
<td>[10, 20, 30]</td>
<td>~356h Train ~63h Validation</td>
</tr>
<tr>
<td>ConvNet 6</td>
<td>[10, 20, 30]</td>
<td>~534h Train ~63h Validation</td>
</tr>
</tbody>
</table>

![Table 2: CDNN systems performance for LID on the 8 languages subset of NIST LRE 2009.](image2)

<table>
<thead>
<tr>
<th>ID</th>
<th>Size</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-vector</td>
<td>~23M</td>
<td>16.94 0.1535</td>
</tr>
<tr>
<td>ConvNet 1</td>
<td>~198k</td>
<td>22.14 0.2406</td>
</tr>
<tr>
<td>ConvNet 2</td>
<td>~39k</td>
<td>25.90 0.2700</td>
</tr>
<tr>
<td>ConvNet 3</td>
<td>~39k</td>
<td>24.69 0.2616</td>
</tr>
<tr>
<td>ConvNet 4</td>
<td>~39k</td>
<td>23.48 0.2461</td>
</tr>
<tr>
<td>ConvNet 5</td>
<td>~78k</td>
<td>21.60 0.2282</td>
</tr>
<tr>
<td>ConvNet 6</td>
<td>~78k</td>
<td>21.11 0.2293</td>
</tr>
<tr>
<td>ConvNet 6+i-vector</td>
<td>-</td>
<td>15.96 0.1433</td>
</tr>
</tbody>
</table>

We train our systems with sequences of 2 s of MFCC-SDCs with no stacking of acoustic frames. They consist of 1 or 2 LSTM hidden layers followed by a softmax output layer, which returns a probability for each input frame and language. For scoring, we average the output per frame, using just the last 10% of each utterance.

#### 2.2.1. Experiments and Results

Table 3 summarizes the performance of 5 LSTM systems in terms of EER$_{avg}$ and accuracy. We see that 4 out of the 5 proposed architectures for the LSTM system outperform the reference i-vector based system in EER$_{avg}$, with 5 to 21 times fewer parameters. In the models with one layer, increasing the number of units improves the performance. However, deeper models (2 layers) yield better results.

### 3. Frame-by-frame DNN-based Representation: Bottleneck Features

Generally, LID and SID systems based on BNF use a DNN with a bottleneck (BN) layer that is trained for ASR. Then, for each input frame, the time-dependent output of the BN layer is used as a new frame-by-frame representation to feed the i-vector model, instead of the classical cepstral features (see Figure 2).

![Figure 2: Scheme of cepstral vs. BN based LID/SID systems.](image3)

Thus, BNFs provide a new frame-wise representation of an audio signal, learned directly by a DNN, containing information about the phonetic content since the DNN is trained for ASR. The BN layer of this DNN is relatively small with respect to the rest and aims to compress the information learned by the previous layers [9].
Table 4: Results of BNFs for LID (development set of NIST LRE 2015), varying the number of layers of the DNN.

<table>
<thead>
<tr>
<th>Number of Hidden Layers</th>
<th>DNN Frame Acc.</th>
<th>EER$_{avg}$ (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>4.64</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5.22</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.87</td>
</tr>
</tbody>
</table>

Table 5: Results of BNFs for LID (development set of NIST LRE 2015) with different position for the BN layer.

<table>
<thead>
<tr>
<th>Position of BN Layer</th>
<th>DNN Frame Accuracy</th>
<th>EER$_{avg}$ (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>49.17</td>
<td>12.24</td>
</tr>
<tr>
<td>Second</td>
<td>49.46</td>
<td>9.55</td>
</tr>
<tr>
<td>Third</td>
<td>49.55</td>
<td>7.81</td>
</tr>
<tr>
<td>Fourth</td>
<td>48.05</td>
<td>8.00</td>
</tr>
</tbody>
</table>

3.1. Analysis of Bottleneck Features for LID

Here, we analyze how the topology of the DNN trained for ASR influences the performance of the resulting BNFs for LID on the NIST LRE 2015 development dataset. We use a feedforward DNN with an input layer, three to five hidden layers, and the output layer. To feed the network, we use 20 MFCCs preprocessed with a context of 31 frames. The hidden layers are composed of 1500 units and the BN layer, of 80. The softmax output layer provides the probability of each input to correspond to a given phoneme state (3083 triphone states are used). We use stochastic gradient descent to optimize the cross-entropy.

3.1.1. Experiments and Results

First we vary the number of layers in the DNN from 3 to 5 (Table 4). Despite the 5 layers configuration gives better performance in terms of frame accuracy, it is the architecture with 4 hidden layers the one that reaches the lowest EER$_{avg}$ for LID. Therefore, the discriminative task (ASR) is easier for the DNN when the classifier is more complex (5 layers DNN) and, thus, improves the frame accuracy. However, that network is not being forced to focus on obtaining a compact representation of the signal, which is then used for LID.

Then, keeping fixed the number of layers to 4, we explore how the LID system performs depending on the position that the BN layer occupies in the DNN, which correspond to different levels of extracted information closer or further from the phonetic information (output layer). Results can be seen in Table 5. The closer the BN layer to the input layer, the noisier the resulting representation would be, which might explain the drop in performance for the first and second layers with respect to the results of the last two layers. The best performance in terms of EER$_{avg}$ for LID is obtained when the bottleneck layer is located in the third layer, but that result is very close to the one obtained with the BN at the fourth layer. Performance of the DNN also drops when the BN layer moves from layer third to fourth. In this topology, the BN layer in position fourth is connected directly to the output layer, resulting in a weight matrix that connects a small layer with just 80 hidden units with the output layer, of size 3083. These weights might be difficult to learn, which may explain this drop in performance of the DNN.

3.2. Analysis of Bottleneck Features for SID

We explore whether DNNs suboptimal for ASR can provide better BNFs for SID. We present here experiments with different features to feed the DNN, either optimized for ASR (“ASR feat.”) or for SID (“MFCC”). The ASR optimized features [10] are composed of 24 Mel-filter bank log outputs concatenated with 13 fundamental frequency (F0) features, with utterance mean normalization, which is what we used as default for ASR [11]. SID optimized features are the classical 20 MFCCs used for SID, either adding the derivatives or not, and normalized with short-term cepstral mean and variance normalization (ST-MVN). We evaluate the systems on the NIST SRE 2010, condition 5, female task [12].

3.2.1. Experiments and Results

The aspect analyzed in this section is the DNN input features, which are either optimized for ASR or SID (“ASR feat.” vs. “MFCC”). Results of these experiments are summarized in Table 6.

We see that the ASR features (with per utterance mean normalization) yield better performance in terms of phone accuracy than the MFCCs since they are expected to be optimized for ASR. However, BNFs obtained from these DNNs do not seem to be as discriminative as the ones obtained with DNNs trained using MFCCs optimized for SID. Moreover, adding first and second derivatives to MFCCs provide better phone accuracy but resulted in a worse SID performance. We see as for LID that better ASR performance (in terms of phone accuracy) does not necessarily correspond to better SID performance.

4. Utterance Level Representation: DNN-based Embeddings

Despite the success of BNFs for SID [13, 14, 15] and LID [16, 17, 18, 19, 20, 21], the variable length of this frame-wise representation poses a challenge in consequent modeling. The classical i-vector compactly the utterance representation in a fixed-length vector. However, the aim of i-vectors is to capture information about sources of variability in the training data, but this information is not necessarily relevant to the target task.

In this section we use embeddings for LID (after their success for SID [22]), which are a fixed-length representation of an utterance extracted from a sequence summarizing DNN trained discriminatively for the target task (LID).

The DNN consists of a first part that works on a frame-by-frame basis from a given sequence of feature vectors, followed by a pooling layer, which in our case computes the mean and standard deviation over time of the activations of the previous layer. Finally, a number of hidden layers follow to capture the information contained in the input, providing a single vector of values per sequence (embeddings), which can be modeled by some other backend.

In particular, our DNN-embedding system takes stacked BNFs as input and use bidirectional LSTM (BLSTM) layers for
the frame-level part. After pooling, two more fully connected layers are added, whose output values will serve as embeddings. Finally, the output layer consists of a softmax layer that provides a vector of language posterior probabilities for each utterance. An example of this architecture is depicted in Figure 3.

4.1. Analysis on NIST LRE 2015

For these experiments, we use the architecture described above. We feed the DNN with 30-dimensional stacked BNFs and the output softmax layer provides a 20-dimensional vector of language posterior probabilities for each utterance. As reference, we use an i-vector system that consists of a 2048-dimensional UBM trained on the same BNFs and 600 dimensional i-vectors.

First, we experiment varying the size of the embedding layers (keeping fixed the rest to 256 for each layer up to the pooling). We start with 512 and 300-dimensional embeddings (DNN_1) and half each twice (DNN_2 and 3, respectively). Results stacking both embeddings are shown in Table 7. We see a better performance of DNN_2 embeddings, which are half size w.r.t. DNN_1. This suggest that the embeddings of larger size contain more detrimental information about channel since all DNNs reached the same performance on the training data.

Motivated by this, we explore further dimensionality reduction via PCA in Table 8. We see that with smaller embeddings, we are able to get improvements even reducing the dimensionality up to 25. Best results are achieved with embeddings from DNN_2 whose dimensionality (406) is close to the typical i-vector (400 or 600). Besides, we achieved a performance of 17.44%, close to our i-vector baseline (16.93%) and score level fusion of both gave us a $C_{avg}$ of 15.69%.

4.2. Analysis on NIST LRE 2017

After developing the embedding system for the NIST LRE 2015, where it was included in the primary submission of BUT team (a fusion of 3 i-vector systems and the embedding system), we performed a post-evaluation analysis.

First, we compared the architecture with the same configuration as DNN_1 in previous section with a larger one, where the fully connected layer has 1500 units and both embeddings are 512-dimensional. With that larger model, performance improved from 22.18% to 19.86% ($C_{primary}$).

Moreover, we extended the training dataset by performing data augmentation through addition of noise, reverberation and tempo variations of original audio files. Figure 4 shows the comparison of performance when training with up to 11 copies of the original data with different corruptions. In general, increasing the number of copies of the data yields improvements in performance. In particular, adding any noisy version of the data (combined or not with other corruptions) makes the system more robust against data mismatch, providing gains in performance. The only two cases in which data augmentation does not improve the system trained only on original data are the ones in which just reverberation or tempo variations are performed.

5. Conclusions

The main contributions of this Ph.D. Thesis are the following. First, the proposed end-to-end approaches for LID based on CDNNs and LSTMs, which provide an alternative to i-vectors with less parameters. Secondly, the systematic study of bottleneck feature DNN-based LID systems and the analysis of this approach for SID, which show that optimal DNN configuration for BNFs for LID and SID might differ from the most beneficial for ASR, task for which the DNN is trained. Finally, the novel approach based on embeddings for LID, in line with previous works in SID, which provides a fixed-length representation of utterances directly learned by the DNN for the target task able to outperform the well-established i-vectors.

In terms of articles, from research directly output from this Ph.D. Thesis, two journal articles and seven peer reviewed international conference papers were published.
6. References


Deep Learning for i-Vector Speaker and Language Recognition: A Ph.D. Thesis Overview

Omid Ghahabi

TALP Research Center, Department of Signal Theory and Communications
Universitat Politecnica de Catalunya - BarcelonaTech, Spain

omid.ghahabi@upc.edu

Abstract

Recent advances in Deep Learning (DL) technology have improved the quality of i-vectors but the DL techniques in use are computationally expensive and need speaker or/and phonetic labels for the background data, which are not easily accessible in practice. On the other hand, the lack of speaker-labeled background data makes a big performance gap, in speaker recognition, between two well-known cosine and PLDA i-vector scoring techniques. This thesis tries to solve the problems above by using the DL technology in different ways, without any need of speaker or phonetic labels. We have proposed an effective DL-based backend for i-vectors which fills 46% of this performance gap, in terms of minDCF, and 79% in combination with a PLDA system with automatically estimated labels. We have also developed an efficient alternative vector representation of speech by keeping the computational cost as low as possible and avoiding phonetic labels. The proposed vectors are referred to as GMM-RBM vectors. Experiments on the core test condition 5 of the NIST SRE 2010 show that comparable results with conventional i-vectors are achieved with a clearly lower computational load in the vector extraction process. Finally, for the LID application, we have proposed a DNN architecture to model effectively the i-vector space of languages in the car environment. It is shown that the proposed DNN architecture outperforms GMM-UBM and i-vector/LDA systems by 37% and 28%, respectively, for short signals 2-3 sec.

Index Terms: Deep Learning, Speaker Recognition, i-Vector, Deep Neural Network, Deep Belief Network, Restricted Boltzmann Machine

1. Introduction

The successful use of Deep Learning (DL) in a large variety of signal processing applications, particularly in speech processing (e.g., [1, 2, 3]), has inspired the community to make use of DL techniques in speaker and language recognition as well. A possible use of DL techniques in speaker recognition is to combine them with the state-of-the-art i-vector [4]. However, the main problem is that the use of DL increases the computational cost of the i-vector extraction process and phonetic and/or speaker labels are required for training, which are not always accessible (e.g., [5, 6, 7, 8, 9]).

Another possible use of DL is to represent a speech signal with a single low dimensional vector using a DL architecture, rather than the traditional i-vector algorithm. These vectors are often referred to as speaker embeddings (e.g., [10, 7, 11, 12, 13]). The need of speaker labels for training the network is one of the disadvantages of these techniques. Moreover, speaker embeddings extracted from hidden layer outputs are not so compatible with Probabilistic Linear Discriminant Analysis (PLDA) backend [14, 15] as the posterior distribution of hidden layer outputs are usually not truly Gaussian.

The first objective in this thesis is to make use of deep architectures for backend i-vector classification in order to fill the performance gap between the cosine (unlabeled-based) and PLDA (labeled-based) scoring baseline systems given unlabeled background data. The second one is to develop an efficient framework for vector representation of speech by keeping the computational cost as low as possible and avoiding speaker and phonetic labels. The last main objective is to make use of deep architectures for backend i-vector classification for Language Identification (LID) in intelligent vehicles. In this scenario, LID systems are evaluated using words or short sentences recorded in cars in four languages, English, Spanish, German, and Finnish.

The three main objectives are summarized in sections 2-4. Section 5 describes the experimental results. Section 6 lists the publications resulted from the Ph.D. thesis and section 7 concludes the paper.

2. Deep Learning Backend for i-Vector Speaker Verification

We have proposed the use of DL as a backend in which a two-class hybrid Deep Belief Network (DBN)-Deep Neural Network (DNN) is trained for each target speaker to increase the discrimination between target i-vector/s and the i-vectors of other speakers (non-targets/impostors) (Fig. 2). Proposed networks are initialized with speaker-specific parameters adapted from a global model, which is referred to as Universal Deep Belief Network (UBBN). Then the cross-entropy between the class labels and the outputs is minimized using the backpropagation algorithm.

DNNs usually need a large number of input samples to be trained efficiently. In speaker recognition, target speakers can be enrolled with only one sample (single session task) or multiple samples (multi-session task). In both cases, the number of target samples is very limited. A network trained with such limited data is highly probable to overfit. On the other hand, the
the cluster centroids are considered as final impostor samples for each target speaker model. Impostor centroids and target samples are then divided equally into minibatches to provide balanced impostor and target data in each minibatch. On the other hand, the DBN adaptation block is proposed to compensate the lack of input data. As DBN training does not need any labeled data, the whole background i-vectors are used to build a global model, which is referred to as Universal DBN (UDBN). The parameters of the UDBN are then adapted to the balanced data obtained for each target speaker. At the end, given the target/impostor labels, the adapted DBN and the balanced data, a DNN is discriminatively trained for each target speaker. More details can be found in [16].

3. RBMs for Vector Representation of Speech

Recently, the advances in DL have improved the quality of i-vectors, but the DL techniques in use are computationally expensive and need phonetic labels for the background data. It has been proposed in this thesis an alternative vector-based representation for speakers in a less computationally expensive manner with no use of any phonetic or speaker labels.

RBM are good potentials for this purpose because they have good representational powers and they are unsupervised and computationally low-cost. It is assumed in this work that the inputs of RBM, i.e., visible units, are GMM supervectors and the outputs, i.e., hidden units, are the low-dimensional vectors we are looking for. The RBM is trained given the background GMM supervectors and will be referred to as URBMs. The role of the URBMs is to learn the total session and speaker variability among the background supervectors. Different types of units and activation functions can be used for training the URBMs but we have proposed a variant of ReLU, which will be referred to as Variable ReLU (VReLU), for this application. It will be shown in section 5 that the proposed VReLU does not suffer from the problems with sigmoid and ReLU and works the best. After training the URBMs, the visible-hidden connection weight matrix is used to transform unseen GMM supervectors to lower dimensional vectors which will be referred to as GMM-RBM vectors in this work.

In fact, the proposed VReLU is defined as follows and is compared with ReLU function in Fig. 3.

$$ f(x) = \begin{cases} x & x > \tau \\ 0 & x \leq \tau \end{cases}, \quad \tau \in N(0,1) \quad (1) $$

Given the GMM supervectors and the URBMs parameters, the GMM-RBM vectors are extracted as follows,

$$ \omega_u = W \Sigma_{ubm}^{-1/2} N^{-1}(u) F(u) \quad (2) $$

where $\Sigma_{ubm}$ is the diagonal covariance matrix of the UBM, $W$ is the connection weights from URBMs, and $N(u)$ and $F(u)$ are zeroth and centralized first order Baum-Welch statistics, respectively.

Like in case of i-vectors, resulting GMM-RBM vectors are mean normalized and whitened using the mean vector and the whitening matrix obtained on the background data.

The comparison of equation 2 with that of i-vector in equation 3 implies clearly that GMM-RBM vector extraction needs much less computational load. More details can be found in [16].

$$ \omega = (I + T^i \Sigma^{-1} N(u) T)^{-1} T^i \Sigma^{-1} F(u) \quad (3) $$
4. Deep Learning Backend for i-Vector Language Identification

Figure 4 shows the architecture of DNNs we have proposed in this work. The inputs are i-vectors and the outputs are the language class posterior. The softmax and sigmoid are used as the activation functions of the internal and the output layers, respectively. In order to Gaussianize the output posterior distributions, we have proposed to compute the output scores in Log Posterior Ratio (LPR) forms as in [16].

As the response time of the LID system is important in the car, the computational complexity of the classifier should also be taken into account. Therefore, we have proposed to choose the size of the first hidden layer as the lowest power of 2 greater than the input layer size. From the second hidden layer towards the output, the size of each layer will be half of the previous layer. For example, the configuration of a 3-hidden-layer DNN will be as 400-512-256-128-4, where 400 is the size of the input i-vectors and 4 is the number of language classes. It will be shown in section 5 that, in this way, we can decrease the computational complexity to a great extent while keeping the classification accuracy.

Two forms of i-vectors are considered as inputs to DNNs, raw i-vectors and session-compensated i-vectors. LDA and WCCN are two commonly used techniques for session variability compensation among i-vectors. Although LDA performs better than WCCN for the LID application when cosine scoring is used, we will use only WCCN session-compensated i-vectors as inputs but no gain was observed. The use of raw i-vectors is advantageous as no language-labeled background data is required. More details can be found in [16].

5. Experimental Results

This section summarizes the main results obtained on the experiments for each main contribution presented in sections 2–4.

The full database provided in the National Institute of Standard and Technology (NIST) 2014 speaker recognition i-vector challenge [17] is used for the experiments in section 2. Rather than speech signals, i-vectors are given directly by NIST in this challenge to train, test, and develop the speaker recognition systems. This enables system comparison more readily with consistency in the front-end and in the amount and type of the background data [17]. Three sets of 600-dimensional i-vectors are provided: development, train, and test consisting of 36,572, 6530, and 9634 i-vectors, respectively. The number of target speaker models is 1306 and for each of them five i-vectors are available. Each target model will be scored against all the test i-vectors and, therefore, the total number of trials will be 12,582,004. Three baseline systems are considered in this work for evaluation: cosine, PLDA with actual labels, and PLDA with estimated labels. The size of hidden layers is set to 400. Table 1 compares the performance of the proposed DNN systems with other baseline systems in terms of minDCF and EER. The interesting point is that the combination of the DNN-1L and PLDA with estimated labels in the score level improves the results to a great extent. The resulting relative improvement compared to cosine baseline system is 36% in terms of minDCF on the evaluation set.

Table 1: Performance comparison of the proposed DNN system with other baseline systems on NIST 2014 i-vector challenge.

<table>
<thead>
<tr>
<th>Unlabeled Background Data</th>
<th>Progress Set</th>
<th>Evaluation Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EER (%)</td>
<td>minDCF</td>
</tr>
<tr>
<td>[1] cosine</td>
<td>4.78</td>
<td>0.386</td>
</tr>
<tr>
<td>[2] PLDA (Estimated Labels)</td>
<td>3.85</td>
<td>0.300</td>
</tr>
<tr>
<td>[3] Proposed DNN-1L</td>
<td>5.13</td>
<td>0.327</td>
</tr>
<tr>
<td>[4] Proposed DNN-3L</td>
<td>4.55</td>
<td>0.305</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Labeled Background Data</th>
<th>Progress Set</th>
<th>Evaluation Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EER (%)</td>
<td>minDCF</td>
</tr>
<tr>
<td>[5] PLDA (Actual Labels)</td>
<td>2.23</td>
<td>0.226</td>
</tr>
<tr>
<td>Fusion [2] &amp; [5]</td>
<td>2.04</td>
<td>0.220</td>
</tr>
<tr>
<td>Fusion [4] &amp; [5]</td>
<td>2.13</td>
<td>0.221</td>
</tr>
</tbody>
</table>
Table 2: Performance comparison of proposed GMM-RBM vectors and conventional i-vectors on the evaluation set core test condition-common 5 of NIST SRE 2010. GMM-RBM vectors and i-vectors are of a same size of 400.

<table>
<thead>
<tr>
<th></th>
<th>cosine</th>
<th>PLDA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EER (%)</td>
<td>minDCF</td>
</tr>
<tr>
<td>[1] i-Vector</td>
<td>6.270</td>
<td>0.05450</td>
</tr>
<tr>
<td>[2] GMM-RBM Vector (Trained with ReLU)</td>
<td>6.638</td>
<td>0.06228</td>
</tr>
<tr>
<td>[3] GMM-RBM Vector (Trained with VReLU)</td>
<td>6.497</td>
<td>0.06099</td>
</tr>
<tr>
<td>Fusion [1] &amp; [3]</td>
<td>5.791</td>
<td>0.05238</td>
</tr>
</tbody>
</table>

Table 3: Comparison of LID systems for short signals recorded in car. Performance values are reported based on LER (%).

<table>
<thead>
<tr>
<th>Duration of Test Signals (in sec)</th>
<th>Number of Samples</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>t &lt; 2</td>
<td>2,472</td>
<td>10,418</td>
</tr>
<tr>
<td>2 ≤ t &lt; 3</td>
<td>2,355</td>
<td>5,591</td>
</tr>
<tr>
<td>t ≥ 3</td>
<td>2,355</td>
<td>5,591</td>
</tr>
</tbody>
</table>

For the experiments in Section 3, the NIST 2010 SRE [18], core test-condition common 5, is used for evaluation. Table 2 compares the performance of GMM-RBM vectors, which are obtained with URBMs trained with ReLU and VReLU, with traditional i-vectors on the evaluation set. The use of proposed VReLU shows better performance than the use of ReLU in both cosine and PLDA scoring. At the end, the best results are achieved with score fusion of i-vectors and GMM-RBM vectors which shows about 7-7.5% and 4-6.5% relative improvements in terms of EER and minDCF, respectively, compared to i-vectors. For score fusion, BOSARIS toolkit [19] is used.

For the experiments of Section 4, the database has been recorded within the scope of the EU project SpeechDat-Car (LE4-8334) [20]. Table 3 summarizes the results for all the techniques in four categories based on the test signal durations: less than 2 sec, between 2 and 3 sec, more than 3 sec, and all durations. The first two categories are more interesting because the decision should be made fast in this application. Both i-vector+DNN systems show superior performance compared to i-vector + LDA baseline system. The frame-based GMM-UBM baseline system works better than other systems only for test signals shorter than 2 sec. However, the accuracy is still high in comparison to other categories.

6. Publications


7. Conclusions

The main contributions of this thesis have been presented in three main works. In the first one, a hybrid architecture based on DBN and DBN has been proposed to discriminatively model each target speaker for i-vector speaker verification. It was shown that the proposed hybrid system fills approximately 46% of the performance gap between the cosine and the oracle PLDA scoring systems in terms of minDCF. In the second work, a new vector representation of speech has been presented for text-independent speaker recognition. Gaussian Mixture Model (GMM) supervectors have been transformed by a Universal RMB (URMB) to lower dimensional vectors, referred to as GMM-RBM vectors. The experimental results show that the performance of GMM-RBM vectors is comparable with that of traditional i-vectors but with much less computational load. In the third work, a DNN architecture has been proposed for i-vector LID of short utterances recorded in cars. It has been shown that for test signals with duration 2-3 sec the proposed DNN architecture outperforms GMM-UBM and i-vector/LDA baseline systems by 37% and 28%, respectively.
8. References


Unsupervised Learning for Expressive Speech Synthesis

Igor Jauk

1Universitat Politècnica de Catalunya, Spain

ij.artium@gmail.com

Abstract

This article describes the homonymous PhD thesis realized at the Universitat Politècnica de Catalunya. The main topic and the goal of the thesis was to research unsupervised manners of training expressive voices for tasks such as audiobook reading. The experiments were conducted on acoustic and semantic domains. In the acoustic domain, the goal was to find a feature set which is suitable to represent expressiveness in speech. The basis for such a set were the i-vectors. The proposed feature set outperformed state-of-the-art sets extracted with OpenSmile.

Involving the semantic domain, the goal was first to predict acoustic features from semantic embeddings of text for expressive speech and to use the predict vectors as acoustic cluster centroids to adapt voices. The result was a system which automatically reads paragraphs with expressive voice and a second system which can be considered as an expressive search engine and leveraged to train voices with specific expressions.

The third experiment evolved to neural network based speech synthesis and the usage of sentiment embeddings. The embeddings were used as an additional input to the synthesis system. The system was evaluated in a preference test showing the success of the approach.

Index Terms: expressive speech, unsupervised training, semantic embeddings, acoustic features

1. Introduction

Speech synthesis is an old, almost romantic, idea of machines, computers, and robots, who talk and express themselves as do human beings. In futuristic science fiction movies and literature, there is almost no way around a talking computer or robot. Sometimes, the talking computer, despite of the vast artificial intelligence capabilities, is identified as such, talking in a robotic and monotonous way, and being nothing else than an aid to humans. In different occasions, computers act very human-like, imitating emotions and free will. In today’s world of smartphones, mobile connectivity and fast-paced life, speech applications gain more and more importance – not least due to ever-improving synthetic speech quality. A fast look at these applications reveals clearly that today’s users do not looking for an AI with a robotic and monotonous voice: making jokes, understanding and showing sympathy, and a long etc. of things which can be resumed in “sounding human-like”, almost seems to be a warrant for product quality of such applications.

In a not too far past, expressive speech (or also emotional speech) was pretty much of an effort. Systems designed specific emotions, often with control mechanisms for intensity etc., and achieved impressive results with such techniques – e.g. [6, 28, 5, 13, 41], just to name a few. However, despite the partly great sounds results, and apart of the laborious design, these systems focus on a limited set of emotions or expressive speaking styles. Real-life human speech does include an infinite number of emotions as those are speaker and situation dependent. So in order to create a truly flexible system you need a mechanism which can adopt to all possible conversational situations and create expressions “on the fly”. For this we need automatic processes for analysis of data and training. Some suggestions to approach this problem were made for instance in [11, 39, 7, 23], where data was clustered by different criteria to select training data and similar ideas. And this is also the topic of this work. The guiding hypothesis are:

1. It is possible to define expressive voices from clusters of data in the acoustic domain, applying unsupervised methods to build the clusters, i.e. no labels of human interpretation are permitted to define the voices or the data in the clusters.
2. It is possible to improve the expressiveness of a synthetic voice using in the training process semantic features which codify some sort of expressive information and are obtained fully automatically.

So one of the main goals is the automatic processing. Also, as the hypothesis reveal, the work embraces both, the acoustic, and the semantic domains. The different approaches are presented in the sections below. Section 2 introduces i-vector-based representation methods for expressive speech. Section 3 introduces prediction methods of acoustic features from semantic text representations, also called embeddings. Section 4 discusses a neural network based TTS system which uses sentimento embeddings to “predict” expressiveness. Finally, Section 5 provides a small discussion and draws some conclusions, and Section 6 lists the helping parties of this project.

2. Acoustic feature selection

Traditionally, expressive speech synthesis and analysis leveraged especially prosodic features to represent emotions or speaking styles. For instance [19] use F0, intensity and duration; [35] use glottal parameters. In [32] the authors use a set of prosodic parameters combined with some spectral ones; [11] use prosodic features, i.e. F0, voicing probability, local jitter and shimmer, and logarithmic HNR for audiobook clustering and posterior synthetic voice training. Almost all research concentrate on prosodic features. However, there are also findings which state that some emotions might be better represented with spectral parameters, e.g. [27, 1]. In [22] the authors used i-vectors (e.g. [31, 8]) to predict emotions. This idea was picked up in [17, 16] and developed to a set of features, i-vector-based and others, which were compared in three clustering experiments.

2.1. Experimental framework

The first and the second experiments compared multiple features objectively and subjectively, as published in [17, 16]. The framework is presented in figure 1.
The idea is: use different feature sets to cluster expressive speech, use the data in clusters to train expressive voices and synthesize a dialogue using these expressive voices. The corpus is a juvenile narrative audiobook recorded in European Spanish, with a total of 7900 sentences and 8.8 hours of duration. The clustering algorithm is k-means, concretely \( VQ \), as by [12]. Many different feature combinations were tested. The results of some of them are presented below. Some features were combined to sets in order to facilitate the notation.

- \( \text{silRate} \) is silence rate and \( \text{sylRate} \) is syllable rate (\#/sec) (extracted with Ogmios [4]).
- \( \text{Rhythm} \) is silence and syllable rates (\#/sec), duration means and variation, computation based on segmentation.
- \( \text{Pitch} \) is \( F_0 \) means, variance and range.
- \( \text{JShimm} \) is Local jitter and shimmer (extracted with [3]).
- \( \text{MFCCiVec} \) are i-vectors calculated on basis of MFCCs;
  - \( \text{F0Vec} \) are i-vectors calculated on basis of \( F_0 \). \( iVecC \);
  - \( F_0 \) and MFCC based i-vectors (the acoustic features for the i-vectors were extracted with the AHOCoder [9], the i-vectors themselves were extracted using Kaldi [29]).

2.2. Objective evaluation

An objective evaluation was performed: a small part of the corpus was labeled with expressions and character (speaker) labels (only for the evaluation purposes) and with the aid of these labels, the perplexity of the clusters was calculated, derived from entropy as by [33, 42]: \( PP = 2^{-H(X)} \).

A resume of the results is shown in the Table 1. Due to space limitations only few chosen results are shown. The upper line shows the perplexity calculated for the annotated part of the corpus. The part below it shows the performance of some “traditional features” and the part at the bottom, the performance of sets which included i-vectors.

As can be seen, the combination of Rhythm and iVecC outperformed all other combinations in the given task.

In order to verify these results, an additional objective evaluation was conducted. For this evaluation, OpenSmile feature sets, as in openSMILE Book [10], were compared to the proposed sets. OpenSmile is a set of feature extraction tools widely used for emotional and expressive speech analysis and synthesis. It extracts thousands of features and statistics about them and is considered to be state-of-the-art feature extraction for expressive speech. The table 2 compares some OpenSmile feature sets to the winning i-vector set: Rhythm & iVecC. Also here, the proposed combination of Rhythm & iVecC outperformed all OpenSmile sets.

2.3. Subjective evaluation

Two subjective experiments were conducted. For these experiments, for a given dialogue from the same audiobook (test set excluded form the training set), a set of synthetic voices was trained using the data in the clusters. The underlying system was an HMM based TTS [24] where the average voice was trained using the whole corpus (approx. 10h) and the cluster data was used to perform adaptation. A total of 16 sentences was presented to a number of participants. The task: design your own audiobook dialogue using synthetic voices instead of the real ones.

The interface is a website, where the participants could choose 1 of 10 synthetic voices for each sentence in a dialogue. The website design aimed to create the right atmosphere of the book story and a more enjoyable experience. Also an introduction text was provided for the case that the participants were not familiar with the story.

The experiments differ in so far, that in the first experiments the participants had an example of the original character voice, in the second they did not. Also in the first experiment, the synthetic voices were chosen manually with the criterion of resemblance to the original voices, and mixed with other random voices. In the second that choices was made automatically by acoustic distance for half of the sentences, the rest was random.

\footnote{\texttt{is09}, \texttt{is10}, \texttt{emobase}, \texttt{emolarge} are feature sets by OpenSmile used in different experiments. For further details please refer to openSMILE Book [10].}
The idea behind: if some voices are especially suitable for some characters, the participants would tend to prefer them.

In the first experiment, 19 persons had participated; in the second, 11 persons. Due to space limitations only the results for the second experiment are shown in Table 3.

Table 3: Relative preferences for the voices v0-v9 over the whole paragraph for the narrator (Narr) and the two present characters (Ch2 and Ch3).

<table>
<thead>
<tr>
<th></th>
<th>v0</th>
<th>v1</th>
<th>v2</th>
<th>v3</th>
<th>v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Narr</td>
<td>0.42</td>
<td>0.06</td>
<td>0.00</td>
<td>0.03</td>
<td>0.04</td>
</tr>
<tr>
<td>Ch2</td>
<td>0.13</td>
<td>0.16</td>
<td>0.14</td>
<td>0.23</td>
<td>0.03</td>
</tr>
<tr>
<td>Ch3</td>
<td>0.18</td>
<td>0.13</td>
<td>0.13</td>
<td>0.51</td>
<td>0.00</td>
</tr>
</tbody>
</table>

The results show that there is an actual preference for some voices atop of others. Of course, not all participants have the same imagination of the book characters, especially if they don’t know the book. So individual preferences are out of scope of this task. But the results show clearly, that the approach as well as the proposed feature sets are suitable for the task. The task itself can be interpreted as a simulation of a real-life application of expressive speech. Further details on the experiments and the results are published in [17, 16, 14].

3. Semantics-to-Acoustics Mapping

When we humans read a text aloud –let’s say we read a good night story to a small child–, we probably will read the story with an expressive voice imitating book characters, their emotions in different situations etc. as to engage the child. Although likely we all would read slightly different, though the “quality” of our reading will possibly be judged by our expressive abilities. The question for this section is: What in the text does provide us the necessary information as to adequately adapt our reading style and can it be taught to a machine?

The key approach is the automatic representation of text. Such representations are often called *embeddings* and there is a large number of techniques to calculate them, like for instance [21, 2, 26, 34]. The basic idea is to represent text in terms of word co-occurrences of defined text units (e.g. sentences, paragraphs, etc.). This way each unit is represented as a co-occurrence vector of its own words. More modern techniques train neural networks to predict a certain feature. Then extract the vector representations from intermediate layers of the network, like [26] and [34].

The assumption is that these representations actually also codify expressive information, especially if the underlying network is trained using an “expressive” criterion (see Section 4). So the task is to convert this vector representation into acoustics.

3.1. Experimental framework

The framework is: in a given text corpus, in this case an audiobook, for each sentence of the text a semantic embedding is calculated. This embedding is then used to predict an acoustic feature vector, concretely from the above experiment, which for its part is the centroid of a data cluster. As in the experiments above, these data clusters are used for adaptation in an HMM-TTS. The performance of the system is evaluated in two subjective experiments. For the embeddings the toolkit *word2vec* [40, 25] was used to calculate the word embeddings; the sentence embeddings were calculated as centroids of the word embeddings in the vector space. The vector space has been trained with the *Wikicorpus* [30].

3.2. Subjective evaluation

The task in the first experiment was to read two book paragraphs automatically predicting expressiveness for each sentences. For each sentence in the paragraph an embedding was calculated. It was used to predict an acoustic feature vector as in Section 2. Two prediction models were compared: a nearest-neighbour classifier and a neural network. Both paragraphs were extracted from different books of the same series, as to preserve characters and the ambience. The expressive readings were also compared to a neutral reading. The task was implemented as a preference test. The participants had the option to choose that two systems performed equally.

A total of 21 persons participated in the experiment. Table 4 shows the results for these experiment for both paragraphs (P1 and P2). The results show a clear preference for the expressive systems.

Table 4: Prediction method preferences by users for the first two tasks. DNN method, nearest neighbor (NN) method, neutral voice.

<table>
<thead>
<tr>
<th></th>
<th>DNN</th>
<th>NN</th>
<th>neutral</th>
<th>DNN</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>0.19</td>
<td>0.43</td>
<td>0.00</td>
<td>0.38</td>
<td>0.05</td>
</tr>
<tr>
<td>P2</td>
<td>0.29</td>
<td>0.14</td>
<td>0.04</td>
<td>0.48</td>
<td>0.05</td>
</tr>
</tbody>
</table>

In a further test the prediction system was used as a “search engine” for expressive training data. For this purpose, a keyword called *seed* was used to predict an acoustic vector, which on its side was used as a cluster centroid for acoustic data and for voice adaptation. For example “Mysterious secret in silent obscurity” was used as seed to find training data for a suspense voice. Other trained emotions were angry, happy and sadness, also a neutral voice. Seven sentences were synthesized with each of these voices and again, a preference test was presented to the 21 participants. The synthesized sentences were chosen trying to reflect their expressive meaning. For example “Finally, the holidays begin!” is supposed to be happy and the expectation was that the participants would choose the happy voice for it. Table 5 presents the results for this experiment.

Table 5: Task 3. Voice preference by users for each sentence.

<table>
<thead>
<tr>
<th></th>
<th>happy</th>
<th>angry</th>
<th>suspense</th>
<th>neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Happy1</td>
<td>0.29</td>
<td>0.38</td>
<td>0.24</td>
<td>0.10</td>
</tr>
<tr>
<td>Happy2</td>
<td>0.52</td>
<td>0.24</td>
<td>0.10</td>
<td>0.14</td>
</tr>
<tr>
<td>Angry1</td>
<td>0.14</td>
<td>0.48</td>
<td>0.24</td>
<td>0.14</td>
</tr>
<tr>
<td>Angry2</td>
<td>0.38</td>
<td>0.43</td>
<td>0.14</td>
<td>0.05</td>
</tr>
<tr>
<td>Suspense</td>
<td>0.00</td>
<td>0.05</td>
<td>0.81</td>
<td>0.14</td>
</tr>
<tr>
<td>Sadness</td>
<td>0.10</td>
<td>0.05</td>
<td>0.43</td>
<td>0.43</td>
</tr>
<tr>
<td>Neutral</td>
<td>0.10</td>
<td>0.05</td>
<td>0.43</td>
<td>0.43</td>
</tr>
</tbody>
</table>

The preferences for voices are pretty clear. It is interesting to remark that happy and angry voices for sometimes exchangeable, the same can be said about the sad and neutral voices in...
their respective contexts. Further details on these experiments can be found in [15, 14].

4. NN-based expressive TTS with sentiment

Looking back at the experiment in the previous chapter there are few spontaneous suggestions which can be made as to develop the approach and improve the results. First, nowadays HMM-based synthesis is almost completely replaced by synthesis based on neural networks. NN-based TTS provides new possibilities of leveraging semantic vectors and avoiding clustering, which is an advantage itself since all data is always taken into account in the training process. For this experiment the DNN-based TTS as described in [36] was used.

The second point is the usage of embeddings which are more suitable to represent expressiveness. The authors in [20, 34] propose the Stanford Sentiment Parser. The system is trained on movie reviews and predicts the positivity or negativity of the sentence.

In previous work, neural network based systems have already been combined with semantic vector input, though not for expressive speech. To name a few, [37] use word embeddings to substitute TOBI and POS tags in RNN-based synthesis achieving significant system improvement. [38] enhance the input to NN-based systems with continuous word embeddings, and also try to substitute the conventional linguistic input by the word embeddings. They do not achieve performance improvement, however, when they use phrase embeddings combined with phonetic context, they do achieve significant improvement in a DNN-based system. [38] enhances word vectors with prosodic information, i.e. updates them, achieving significant improvements.

4.1. Experimental framework

The DNN-based system was trained on two audiobooks in American English. An additional linguistic input is introduced, the sentiment predicted by the Stanford sentiment parser. Here, different input combinations are tested, including word context.

With word context, probability vector of the word in question and the probability vectors of two words on the left and two words on the right were used. Also the tree distance, which is the hierarchical distance counted in the number of binary tree nodes between words is added, such that the input vector for each word for the system (\(v_{wcd}\)) is composed as follows:

\[
P = \{P_{2}, P_{1}, P_{c}, P_{r_{1}}, P_{r_{2}}, D\}
\]

where \(P_{2}\) is the probability vector for the current word, the \(P_{1}\) is the probability vector for the second word on the left, \(P_{c}\) is probability vector for the first word on the left, \(P_{r_{1}}\) is probability vector for the first word on the right and \(P_{r_{2}}\) is the probability vector for the second word on the right, each of the probability vectors as defined in equation 2. \(D\) is the hierarchical tree distance (distance in tree counted in nodes).

4.2. Subjective experiments

Two experiments have been conducted in order to test the system performance. In all of them, similar to the experiment above, a preference test is conducted among a group of participants. A total of 20 persons participated in the experiment. However, there was a larger group of speech technology experts and people who have no experience with speech technology which allowed for an interesting comparison of the deviations of these two groups. Due to space limitations only the general preference results will be shown here. More details can be found in [14, 18].

Table 6 shows the preferences divided by the sentiment. For positive and negative sentences, the word level system performed best, although for negative sentences with high variance. For neutral sentences, the word context and tree distance system performed best. Possibly it is due to the fact that it probably has an equilibrating effect.

T-tests show that for negative sentences, there is a significant difference between the system without sentiment and the word level system, and no significant difference for the other systems. For neutral sentences, there is a significant difference between the system without sentiment and the word context and tree distance system, but not for the other systems. For positive sentences, there is only significant difference for the one-tailed t-test between the system without sentiment and the word level system.

<table>
<thead>
<tr>
<th>Sentiment</th>
<th>ws</th>
<th>wcd</th>
<th>wl</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive</td>
<td>1.84</td>
<td>1.85</td>
<td>1.71</td>
</tr>
<tr>
<td>positive variance</td>
<td>0.54</td>
<td>0.76</td>
<td>0.54</td>
</tr>
<tr>
<td>negative</td>
<td>2.06</td>
<td>1.96</td>
<td>1.84</td>
</tr>
<tr>
<td>negative variance</td>
<td>0.52</td>
<td>0.67</td>
<td>1.1</td>
</tr>
<tr>
<td>neutral</td>
<td>2</td>
<td>1.83</td>
<td>1.96</td>
</tr>
<tr>
<td>neutral variance</td>
<td>0.71</td>
<td>0.6</td>
<td>0.95</td>
</tr>
</tbody>
</table>

5. Discussion & Conclusions

The main topic addressed in this thesis is the automatic training of expressive voices. Several experiments were conducted on the acoustic domain, automatically selecting training data, and on semantic domain, predicting acoustics from semantics. In the last experiments, two state-of-the-art techniques were combined for the given task. Speech technology domain is an incredibly fast evolving field, especially being data driven where nowadays data is the key to all information technology. Nevertheless, the underlying technologies leveraged and presented in this work are not only not out of date, but are actually the driving power of current technologies. This includes the NN-based TTS and the semantic embeddings for text representation, but also not least i-vector-like representations for the acoustic domain. In that sense, this thesis is a substantial contribution to speech technology research.

6. Acknowledgements

I would like to thank especially Antonio Bonafonte, the tutor of this thesis. This work was supported by the Spanish Ministerio de Economía y Competitividad and European Regional Development Fund, contract TEC2015-69266-P (MINECO/FEDER, UE) and by the FPU grant (Formación de Profesorado Universitario) from the Spanish Ministry of Science and Innovation (MCINN) to Igor Jauk, with a special collaboration with the University of Texas at El Paso, under Pr. Nigel Ward, and the National Institute of Informatics in Tokyo, under Pr. Junichi Yamagishi.
7. References


[40] word2vec Tool for computing continuous distributed representations of words.


ODESSA/PLUMCOT at Albayzin Multimodal Diarization Challenge 2018

Benjamin MAURICE1, Hervé BREDIN1, Ruiqing YIN1, Jose PATINO3, Héctor DELGADO3, Claude BARRAS2, Nicholas EVANS3, Camille GUINAUDEAU2

1 LIMSI, CNRS, Université Paris-Saclay, Orsay, France
2 LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, Orsay, France
3 EURECOM, Sophia-Antipolis, France

1firstname.lastname@limsi.fr, 2firstname.lastname@eurecom.fr

Abstract

This paper describes ODESSA and PLUMCOT submissions to Albayzin Multimodal Diarization Challenge 2018. Given a list of people to recognize (alongside image and short video samples of those people), the task consists in jointly answering the two questions “who speaks when?” and “who appears when?”. Both consortia submitted 3 runs (1 primary and 2 contrastive) based on the same underlying monomodal neural technologies: neural speaker segmentation, neural speaker embeddings, neural face embeddings, and neural talking-face detection. Our submissions aim at showing that face clustering and recognition can (hopefully) help to improve speaker diarization.

Index Terms: multimodal speaker diarization, face clustering

1. Introduction

This paper describes ODESSA1 and PLUMCOT2 submissions to Albayzin Multimodal Diarization Challenge 2018 [1]. Given a collection of broadcast news TV recordings and a list of people to recognize (alongside image and short video samples of those people), the task consists in jointly answering the two questions “who speaks when?” and “who appears when?”.

While ODESSA submissions are made of the simple juxta-position of two monomodal systems (audio-only speaker diarization on one side, visual-only face recognition on the other side), PLUMCOT runs aimed at showing that face clustering and recognition can help to improve speaker diarization (and vice-versa). Figure 1 provides an overview of PLUMCOT multimodal pipelines. The upper part corresponds to the face recognition pipeline described in details in Section 2. The lower part corresponds to the speaker diarization pipeline further described in Section 3. Section 4 describes the proposed multimodal approaches, depicted by vertical arrows between audio and visual modalities in Figure 1.

Instead of optimizing audio and visual pipelines separately, we propose to tune the whole set of hyper-parameters jointly with respect to the official evaluation metric. This is described in Section 5. Following sections 7 and 8 introduce the experimental protocol and results on the development set3.

2. Face clustering and recognition

This section describes the building blocks of the “face” part of our runs. They all rely on the pyannote.video toolkit introduced in [2]. It mostly consists of three sub-modules: face tracking, neural face embedding, and face clustering.

2.1. Face tracking

After an initial step of shot boundary detection using optical flow and displaced frame difference [3], face tracking-by-detection is applied within each shot using a detector based on histogram of oriented gradients [4] and the correlation tracker proposed in [5]. More precisely, face detection is applied every frame (so every 40ms), and tracking is performed in both forward and backward directions.

2.2. Face embedding

Each face track is then processed using the ResNet network with 29 convolutional layers [6] available in the dlib machine learning toolkit [7]. This network was trained on both FaceScrub [8] and VGG-Face [9] datasets to project each face into a 128-dimensional Euclidean space, in which faces from the same person are expected to be close to each other. Each face track is described by its average face embedding \( x_{face} \).

2.3. Face clustering

Face tracks are grouped together using agglomerative clustering. Clustering is initialized with one cluster per face track belonging to clusters \( i \) and \( j \) respectively. Then, the following process is repeated iteratively until a stopping criterion is reached:

- find the two most similar clusters \( (i, j) \) according to the Euclidean distance \( d_{ij} = d(x_i, x_j) \) between their embedding \( x_i \) and \( x_j \);
- compute the embedding \( x \) of the newly formed cluster as the weighted average of the embedding \( x_i \) and \( x_j \) of the two merged clusters \( i \) and \( j \):

\[
\begin{align*}
x &= \frac{n_i \cdot x_i + n_j \cdot x_j}{n_i + n_j} \\
n &= n_i + n_j
\end{align*}
\]

(1)

(2)

where \( n_i \) and \( n_j \) are the total number of face tracks belonging to clusters \( i \) and \( j \) respectively. This agglomerative process stops when \( d_{ij} \) is greater than a tunable threshold \( \theta_0 \).

2.4. Face recognition

While a perfect face clustering should lead to a perfect (visual) diarization error rate, the actual metric used in the Albayzin Multimodal Diarization Challenge assumes that only a limited list of \( T \) target persons should be returned by the system. Enrollment data is provided for each target, containing approximately 10 pictures and one short video sample showing their face.
Figure 1: Pipeline used for PLUMCOT submissions (p = primary, c = contrastive). $\theta_s$ and $\lambda_s$ are jointly-optimized hyper-parameters.

3. Speaker diarization

This section describes the building blocks of the “speaker” part of PLUMCOT runs. They all rely on the speaker diarization approach introduced in [10].

3.1. Speech turn segmentation

All submission share a same speech activity detection (SAD) system proposed in [11]. SAD is modeled as a supervised binary classification task (speech vs. non-speech), and addressed as frame-wise sequence labeling tasks using bi-directional LSTM on top of MFCC features. For SCD, two systems were explored: the first one named uniform segmentation which splits the speech parts into 1s segments, the second one using the system proposed by [12]. Similar to SAD, SCD is modeled as a supervised binary sequence labeling task (change vs. non-change).

3.2. Speaker embedding

The embedding architecture used is the one introduced in [2] and further improved in [13]. In the embedding space, using the triplet loss paradigm, two sequences $x_i$ and $x_j$ of the same speaker (resp. two different speakers) are expected to be close to (resp. far from) each other according to their angular distance. The embeddings are trained on the Voxceleb corpus.

3.3. Speech turn clustering

As proposed in [10], we use Affinity propagation (AP) algorithm [14] to perform clustering of speech turns. AP does not require a prior choice of the number of clusters contrary to other clustering methods. All speech segments are potential cluster centers (exemplars). Taking as input the pair-wise similarities between all pairs of speech segments, AP will select the exemplars and associate all other speech segments to an exemplar. In our case, the similarity between $i^{th}$ and $j^{th}$ speech segments is the negative angular distance between their embeddings. AP has two hyper-parameters: preference $\theta_{sc}$ and damping factor $\lambda_{sc}$.

3.4. Re-segmentation

A final re-segmentation step is performed to refine time boundaries of the segments generated in the clustering step. It uses Gaussian mixture models (GMM) to model the clusters, and maximum likelihood scoring at feature level. Since the log-likelihoods at frame level are noisy, an average smoothing within a 1s sliding window is applied to the log-likelihood curves obtained with each cluster GMM. Then, each frame is assigned to the cluster which provides the highest smoothed log-likelihood.
4. Multimodal fusion

This section describes our attempts at improving speaker diarization with face clustering, and vice versa. Those two approaches were respectively submitted as PLUMCOT primary run (4.1) and PLUMCOT first contrastive run (4.2).

4.1. Improving speaker diarization with face clustering

Let us assume that there are $N$ speakers according to speaker diarization, and $M$ persons according to face clustering (or recognition). Let $K \in \mathbb{R}^{N \times M}$ be the co-occurrence matrix of the output of both pipelines: $K_{ij}$ is the overall duration in which speaker $i \in \{1 \ldots N\}$ is speaking and person $j \in \{1 \ldots M\}$ is visible.

The main intuition motivating this approach arises from the following observation about broadcast news videos: most of the time, the camera is pointing at the current speaker. Therefore, the proposed approach simply updates each speaker cluster by assigning them to the most co-occurring face cluster:

$$i \leftarrow \mathop{\arg\max}_{i \in \{1 \ldots M\}} K_{ij}$$  \hspace{1cm} (4)

Thanks to the joint optimization (described in Section 5) of stopping criteria for both face clustering and speaker diarization, we anticipate that this approach will “choose” to favour smaller (but purer) speaker clusters than the purely monomodal speaker diarization pipeline. A speaker divided into several small clusters may then be merged back together thanks to (a hopefully better) face clustering and Equation 4.

4.2. Filtering face detection with speech activity detection

![Diagram of speech turns and face tracks](image)

Figure 2: Face tracks within long non-speech regions (red) are removed.

Our face detection and tracking module tends to detect lots of non-target faces, leading to a huge amount of false alarms (e.g. in crowds, in credits at the end of TV shows, etc.). As depicted in Figure 2, we propose a very simple solution to this problem: filtering face tracks in long non-speech regions.

5. Hyper-parameters joint optimization

As mentioned in Section 4.1, the various modules of PLUMCOT runs are jointly optimized. For instance, the “speaker” part of PLUMCOT primary run is the combination of two modules with their own set of hyper-parameters: face clustering ($\theta_{fc}$) and speaker diarization ($\theta_{sc}$ and $\lambda_{sc}$). Instead of tuning the former for optimal face clustering performance and the latter for optimal speaker diarization separately, the whole pipeline (including the assignment step described in Equation 4) is jointly optimized.

Practically, we use the Covariance Matrix Adaptation Evolution Strategy minimization method [15] available in the chocolate library\(^4\) to automatically select the set of hyper-parameters that minimizes the speaker diarization error rate for “speaker” part and the face diarization error rate for “face” part.

6. Submissions

Figure 1 summarizes the primary and two contrastive runs of the PLUMCOT consortium. All of them have been introduced in the previous sections of this paper.

The ODESSA consortium mostly focused on the monomodal speaker diarization aspects of the task. Therefore, ODESSA submissions to the “speaker” part of the multimodal diarization challenge rely on the same systems used for its open-set submissions to the speaker diarization challenge: the fusion at similarity-level of various speech turn representations (such as neural embeddings and binary keys). More information can be found in [16]. All three ODESSA submissions use the same “face” part as PLUMCOT primary submission.

7. Experimental protocol

7.1. RTVE2018 corpus

The RTVE2018 dataset is a collection of diverse TV shows aired between 2015 and 2018 on the public Spanish National Television (RTVE). The development subset of the RTVE2018 database contains one single 2 hours show “La noche en 24H” labeled with speaker and face timestamps. It also contains 11 additional files (for a total duration of 14 hours) labeled with speaker timestamps only. Enrollment files for the target persons are also provided: they consist of a few pictures and one short video for each target.

The evaluation set contains 3 videos files of almost 2 hours each of TV shows labeled with speaker and face timestamps. However, at the time of the submission of this paper, we have no result on the test set so we are not reporting results on this test set.

7.2. Evaluation metric

The evaluation metric used for this task is the diarization error rate (DER) defined as follows:

$$\text{DER} = \frac{\text{false alarm} + \text{missed detection} + \text{confusion}}{\text{total}}$$  \hspace{1cm} (5)

where false alarm is the duration of non-speech incorrectly classified as speech, missed detection is the duration of speech incorrectly classified as non-speech, confusion is the duration of speaker confusion, and total is the total duration of speech in the reference. Note that this metric does take overlapping speech into account, potentially leading to increased missed detection in case the speaker diarization system does not include an overlapping speech detection module. DER is a standard metric for evaluating and comparing speaker diarization systems but it can also be applied for face clustering by replacing speech turns by face tracks.

7.3. Implementation details

7.3.1. Face clustering and recognition

As already stated in Section 2, we use the pre-trained face detector and face embedding available in dlib library [7], wrapped in our pyannote.video toolkit\(^5\). All hyper-parameters of the face

---

\(^4\)chocolate.readthedocs.io

\(^5\)github.com/pyannote/pyannote-video
clustering and recognition pipeline are jointly optimized in order to minimize the (face) diarization error rate on the only annotated video of the RTVE2018 development set provided by the organizers of the challenge.

7.3.2. Speaker diarization

Feature extraction. All modules in the speaker diarization pipeline share the same feature extraction step: 19 MFCC coefficients (with their first and second derivatives, and the first and second derivatives of the energy) are extracted every 10ms on a 25ms windows. The only exception is the re-segmentation step that does not use any derivative.

Segmentation. Both speech activity and speaker change detection modules are trained with the Catalan broadcast news database from the 3/24 TV channel proposed for the 2010 Albayzin Audio Segmentation Evaluation [17]. We use the exact same configuration as the one described in [10]: stacked bi-directional LSTMs and multi-layer perceptron on 3.2s sliding windows.

Speaker embedding. Speaker embeddings are trained using VoxCeleb1 dataset [18]. We use the exact same architecture as the one used in [13] (stacked bi-directional LSTMs on a 3s window) and the training process introduced in [19] (triplet loss with angular distance).

Speaker diarization pipeline. Once every module is trained, hyper-parameters of the speaker diarization pipeline are jointly optimized in order to minimize the diarization error rate on the development set (dev2) of RTVE2018 corpus provided by the organizers of the challenge.

8. Results and discussion

Table 1 summarizes the performance of each submission on the development set. Official results on the test set were not available at the time of writing the paper.

<table>
<thead>
<tr>
<th>Consortium</th>
<th>Run</th>
<th>Speaker</th>
<th>Face</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLUMCOT</td>
<td>primary</td>
<td>6.86</td>
<td>28.15</td>
</tr>
<tr>
<td></td>
<td>contrastive 1</td>
<td>10.59</td>
<td>28.15</td>
</tr>
<tr>
<td></td>
<td>contrastive 2</td>
<td>10.68</td>
<td>31.01</td>
</tr>
<tr>
<td>ODESSA</td>
<td>primary</td>
<td>7.21</td>
<td>28.15</td>
</tr>
<tr>
<td></td>
<td>contrastive 1</td>
<td>9.29</td>
<td>28.15</td>
</tr>
<tr>
<td></td>
<td>contrastive 2</td>
<td>11.46</td>
<td>28.15</td>
</tr>
</tbody>
</table>

Table 1: Diarization error rate on the development set

Comparing “speaker” parts of PLUMCOT primary run (DER = 6.86%) and contrastive run #1 (DER = 10.59%) shows that speaker diarization can be greatly improved when guided by face clustering: this amounts to a relative improvement of 35%. Face clustering also helps significantly for face recognition: it is improved from DER = 31.01% for track-wise face recognition ($c^2_{face}$) to DER = 28.15% for cluster-wise face recognition ($p_{face}$).

There are no difference between face primary run and contrastive run #1 maybe because during the long silence founded faces were already deleted with the recognition threshold $\theta_{fs}$ with the enrollment data.

While cluster-wise face recognition (DER = 28.15%, $p_{face}$) is better than raw face clustering (DER = 46.02%, not shown in Table 1) for the “face” part, the latter does lead to better “speaker” performance than the former when jointly optimized with the speaker diarization pipeline ($p_{speaker}$ gets DER = 6.86% while $c^2_{speaker}$ only gets DER = 10.68%). This shows the benefit of the joint optimization of hyper-parameters: a better “face” system does not necessarily lead to a better multimodal “speaker” pipeline.

As described in details in [16], ODESSA “speaker” primary run is the combination at similarity level of three different representations (x-vector trained on NIST SRE data, triplet loss embedding trained on VoxCeleb and binary key). This complex system reaches a performance of DER = 7.21% which is still below the simpler multimodal PLUMCOT primary run (that combines triplet loss speaker embedding and neural face embedding) with DER = 6.86%. One could hope that combining both approaches would help us get even closer to perfect diarization.

9. Conclusion and future work

We have conducted experiments on monomodal face clustering and speaker diarization and shown an improvement of the results when we combine them into a multimodal approach. It has also been shown that combining two monomodal approaches tuned separately does not automatically lead to the best results: one should rather tune them jointly using a global optimization process.

While results of the multimodal approaches are promising, there is still room for improvement. In particular, we plan to investigate the use of the talking-face detection approach introduced in [20] to improve the module in charge of mapping face clusters with speaker clusters.

Finally, we would like to highlight the fact that the code for most monomodal building blocks is available for other researchers to use\textsuperscript{67}.

10. Acknowledgements

This work was partly supported by ANR through the ODESSA (ANR-15-CE39-0010) and PLUMCOT (ANR-16-CE92-0025) projects.

11. References


\textsuperscript{6}github.com/pyannote/pyannote-audio
\textsuperscript{7}github.com/pyannote/pyannote-video


UPC Multimodal Speaker Diarization System for the 2018 Albayzin Challenge

Miquel India, Itziar Sagastiberri, Ponç Palau, Elisa Sayrol, Josep Ramon Morros, Javier Hernando

Universitat Politècnica de Catalunya (UPC)
miquel.india@tsc.upc.edu, itziar.sf@opendeusto.es, ppalaupigdevall@gmail.com,
elisa.sayrol, ramon.morros, javier.hernando}@upc.edu

Abstract

This paper presents the UPC system proposed for the Multimodal Speaker Diarization task of the 2018 Albayzin Challenge. This approach works by processing individually the speech and the image signal. In the speech domain, speaker diarization is performed using identity embeddings created by a triplet loss DNN\(^1\) that uses i-vectors as input. The triplet DNN is trained with an additional regularization loss that minimizes the variance of both positive and negative distances. A sliding window is then used to compare speech segments with enrollment speaker targets using cosine distance between the embeddings. To detect identities from the face modality, a face detector is followed by a face tracker that has been used on the videos. For each cropped face a feature vector is obtained using a Deep Neural Network based on the ResNet 34 architecture, trained using a metric learning triplet loss (available from dlib library). For each track the face feature vector is obtained by averaging the features obtained for each one of the frames of that track. Then, this feature vector is compared with the features extracted from the images of the enrollment identities. The proposed system is evaluated on the RTVE2018 database.

Index Terms: Speaker Diarization, Face Diarization, Multimodal System

1. Introduction

Video broadcasting generates a massive amount of multimedia information that once archived generates a need to access its contents. In particular, it is essential to develop tools that are able to automatically search and detect the presence of people. Challenges such as REPERE [1] or the MediaEval Multimodal Person Discovery in Broadcast TV [2], [3] addressed the identification of people appearing and speaking.

Two main approaches can be found in the literature for retrieval of person identification in videos: Perhaps the most popular is based on clustering face tracks, speech segments or both [4, 5]. This provides multiple clusters (agglomerations of signal segments) that correspond to the identities in the video. Then, an assignment of names to clusters is performed. The main problem of this approach is that a large number of non important identities can appear, and that the clusters are highly non-homogeneous. Combining speech and face modalities is also a challenge in these systems. The second usual approach is verification on the signal segments [4, 6]. Here, two stages are defined: enrollment and search. Each segment is compared against the enrollment data and a decision is made for each segment. In several cases, both approaches (clustering and verification) use of some kind of metric learning to improve the discriminativeness of the feature vectors.

This paper presents the UPC proposal to the Albayzin Evaluation: IberSPEECH-RTVE 2018 Speaker Diarization Challenge. In this challenge a list of people occurrences within the RTVE2018 database should be provided as a result. This list must contain people either if they are talking, or their faces appear in the video, or both at the same time. The two modalities should therefore be provided. Identities and enrollment data are given, so the identification might be considered supervised. Nevertheless, in the evaluation data unknown persons may appear and should be distinguished from those that are known.

This paper is organized as follows: Section 2 describes the system that has been developed to detect identities from the image and the speech modalities. Section 3 gives the technical details of the setup and the experimental results on the provided database. Finally, conclusions and final remarks are given in section 4.

2. System Description

The UPC Multimodal Speaker Diarization System consists on two monomodal systems and a fusion block that combines the outputs of the previous systems so as to obtain a more refined speaker labelling. The speech and the image are processed in an independent manner with a speaker and a face diarization system. The face tracks and the speaker segments are then fused with an algorithm that combines the intersection of these sources according to a set of assumptions made on the data. The next section describes in detail the monomodal approaches and the fusion system to combine them.

2.1. Video System

The video system is responsible of localizing the faces of the individuals appearing in the scene and to determine if these faces belong to one of the \(N\) given identities.

Our approach is based on performing face tracking to identify the intervals where a given person is appearing in the video. A face track consists of the location of the faces of an individual in the consecutive frames where he/she appears in the video. Thus, the face track determines the spatial location of the faces and the temporal interval in the video where this person appears. Then, each face track is forwarded through a classifier with \(N + 1\) classes, namely the identities of the known persons (this is, the set of persons in the enrollment set) plus the unknown class. The approach is summarized in Figure 1.

\[\text{Figure 1: Block diagram for the face modality.}\]
Our approach uses a tracking by detection approach: First, all the faces in the video sequence are detected. For this, we use a detector based on HOG+SVM\(^2\) [7] from the dlib [8] library.

Once the faces have been detected, a KLT\(^3\) tracker [9, 10, 11] is used to relate the detections in successive frames. We used the implementation\(^4\) provided for the baseline system of the Multimodal Person Discovery in Broadcast TV task in MediaEval 2015 [2].

As mentioned previously, a face track provides the spatial location of a set of faces of a given individual, which are used for feature extraction, and the temporal interval where this person is visible in the video.

In the video system, the track is the basic unit of recognition: we will output a result for each track that is classified as belonging to one of the known persons. Tracks classified as unknown are discarded and no output is provided.

To characterize each track, we follow a two step process: first, a feature vector is extracted for each detected face in the track. Then, the final feature vector for the track is obtained by averaging all the track’s feature vectors.

These feature vectors are obtained using the last fully connected layer from a Deep Neural Network based on the ResNet 34 architecture [12], trained using the metric learning triplet loss process described in FaceNet [13]. This learns a mapping from the detected faces to a compact space where the feature vectors (i.e. 128 dimensional FaceNet embeddings) originating from the faces of a given individual are located in a separate and compact region of the space. Thus, the vectors are highly discriminative, allowing to use standard techniques to perform classification/verification. We have used the off-the-shelf dlib [8] implementation, without any adaptation nor fine-tuning to the task identities.

A similar method is used to extract the feature vectors for the images and videos of the enrollment set. For each person, 10 still images and one short clip were provided. For each still image, we detect the face and we extract a feature vector. The short video is processed similarly to the test video: scene detection, face detection and face tracking. A feature vector is extracted for each resulting track. This results in a variable number of enrollment vectors for each person, depending on the number of tracks in the short video. These vectors are associated with the name of the corresponding person and used as a person model.

To decide the track identity, we used a k-NN classifier with a cosine distance metric. A global threshold is applied to determine if the track belongs to any of the persons in the database. If this is the case, the identity corresponding to the nearest vector in the database is used as the track identity. This simple method is possible because the highly discriminative properties of the FaceNet 128 dimensional embeddings.

### 2.2. Speaker System

The speaker system works as a tracking algorithm that uses speaker embeddings to compare speech segments with the speech utterances of the enrollment identities. These representations are created with a DNN which is feed with i-vectors and is trained with a multi-objective loss (Figure 2). This loss is based on a triplet margin loss and a regularization function which minimizes the variance of both positive and negative tuple distances.

\(^2\) Histogram of Oriented Gradients + Support Vector Machine

\(^3\) Kanade-Lucas-Tomasi tracker

\(^4\) https://github.com/MediaevalPersonDiscoveryTask/Baseline2015

\[ M = \mu + T\omega \] (1)

where \(\mu\) is the speaker- and session-independent mean supervector from UBM, \(T\) is the total variability matrix, and \(\omega\) is a hidden variable. The mean of the posterior distribution of \(\omega\) is referred to as i-vector. This posterior distribution is conditioned on the Baum-Welch statistics of the given speech utterance.

The \( T \) matrix is trained using the Expectation-Maximization (EM) algorithm given the centralized Baum-Welch statistics from background speech utterances. More details can be found in [14].

Given an i-vector, a DNN is used to extract a more discriminative speaker vector. This DNN is composed by 2 hidden layers of 400 nodes, where the activations of the second layer are used as a speaker embedding. This neural network is fed with i-vectors and an initial L2 normalization is applied to these inputs before the first hidden layer. After each hidden layer, a batch normalization layer is used as regularizer. Initially, the DNN is pretrained as a speaker classifier. Therefore, a softmax layer is added in the output of the network and the DNN is trained minimizing the cross-entropy loss. Following to this pretraining, the softmax layer is removed and the DNN is trained with the following multiple objective loss:

\[ \text{Loss} = \frac{1}{N} \sum_{i=1}^{N} T\text{Loss}_i + \frac{\lambda}{2} R\text{Loss}_i \] (2)

\[ T\text{Loss}_i = \max(0, d(A_i, P_i) - d(A_i, N_i) + \text{margin}) \] (3)

![Speaker front-end diagram.](image-url)
Figure 3: Fusion scheme. Green boxes refer to the segments where id assignation has been directly propagated. Orange boxes refer to segments which id ask has been assigned after the score combination.

\[ RLoss_i = (|d(A_i, P_i)| - \frac{1}{N} \sum_{j=1}^{N} |d(A_j, P_j)|) + \\
(\frac{1}{N} \sum_{j=1}^{N} |d(A_j, N_j)|) \]

where \( TLoss \) corresponds to the triplet margin loss [13] and \( RLoss \) corresponds to our proposed regularization function. \( TLoss \) is computed with hinge loss and \( d(x,y) \) refers to the 2\( \times \)2 order euclidean distance between a pair of vectors. On the other hand, \( RLoss \) is a function that forces the DNN to minimize the variance of both positive and negative tuples distances. Hence, in each batch we estimate the means of the positive and negative scores. These means are then used to minimize the distance between each positive or negative pair distance and its corresponding mean. We add a \( \lambda \) penalty term so as to balance the magnitude of the regularization function in the global loss.

The i-vector framework combined with the DNN is used as a front-end block in the speaker tracking system. This front-end allows to extract features of the speech signal and compare it with the signals of the speaker targets. In our approach, a sliding window strategy have been used to extract speaker embeddings from 3 seconds length speech segments with a 0.25 seconds shift size. For the enrollment identities, we have used the whole signal to extract an embedding for each target. Cosine distance metric is then used to evaluate the similarity of speech segment embeddings for each target. The target with the biggest similarity is then assigned to the corresponding speech segment. In order to classify the non-interest or unknown speakers, a threshold is imposed to determine the assignation between the best candidate and the speech segment. If the most similar target distance is below the threshold, the speech segment is automatically tagged as an unknown identity.

### 2.3. Fusion System

A fusion system has been considered in order to combine the previous information sources. Speaker and video diarization are performed first in an individual manner. The results of both modalities are then fused so as to obtain a better speaker assignation. In order to combine both outputs properly, we made the following assumptions:

- Some speakers do not come into view any time in the show and there are other people who are shown in the screen but do not speak. These faces and speakers correspond in major part to the unknown identities.

According to these assumptions, an algorithm has been designed based in weighting temporal overlaps between the tracks of the face system and the speech segments of the speaker system (Figure 3). As its shown in the figure, the intersection between face tracks and speaker segments produces a new multimodal segmentation. The temporary segments where face and speech are not overlapped are discarded. We use this new segmentation to combine the assignations of both modalities:

- The segments where the corresponding face/speaker segments have the same target assigned are automatically tagged with that identity.

- When the speaker and face assignations are not the same, we produce a new scoring combining both modalities distances between the segment and the enrollment targets. First we extract the scores of the multimodal segment for each modality. The range of these scores are different for each source, hence it is needed to normalize them. This normalization is produced with a softmax activation which has a different temperature \( \tau \) parameter for each modality. A new set of scores is then produced with the average of both modalities scores. Given these new multimodal scores, a new threshold is used to determine whether the segment correspond to the most similar target or to an unknown identity.

### 3. Optimization and Experimental Results

In the following section we describe the setup of the proposed approaches and we present the results of these systems for the Multimodal Speaker Diarization task of the 2018 Albayzin Challenge.

#### 3.1. Speaker System

The speaker front-end block has been trained on the Vox-Celeb2 [15] database. Feature extraction is performed with 20 size MFCC plus delta features. The UBM has been trained with a 1024 mixtures GMM and the T Matrix size is 400. For the whole i-vector framework we have used the Alize [16, 17] toolkit and we have only used the first 1000 speakers of Vox-Celeb2 development partition. The DNN used is composed by two 400 size hidden layers. The pretraining has been performed using the same data used for the i-vector framework. For the triplet based DNN training, the whole VoxCeleb2 development partition have been used. In order to obtain a good estimation of the positive and negative pair means, batch size have been set to 1024. The \( \lambda \) for the \( RLoss \) have been set to 1. Both network trainnings have been performed with Adam optimizer. Learning rate have been set to 0.01 and the pretraining has been regularized with an additional 0.001 weight decay. For the target assignation, the decision threshold has been tuned to improve DER results on the RTVE2018 development set. A final value of 0.08 threshold over the the cosine distance (in range [-1,1]) has been obtained.

#### 3.2. Video System

The method described in Section 2.1 has been used to obtain the results. We have filtered short tracks (tracks shorter than 1s) because they are likely to belong to non-important faces.
This also allows to reduce the computational load of the system. For each track, a 128D feature vector has been generated. The final identity decision is determined by a k-NN classifier. As the number of enrollment vectors is low, a value of \( k = 1 \) has been used. By looking at the small Speaker Error Rate value in Table 1, this approach is effective, thanks to the discriminating power of the embeddings. The principal challenge in this task was the high number of tracks belonging to persons that are not in the enrollment set. To reject these tracks, a global threshold \( th \) has been used. This threshold has been determined as the value providing the highest DER measure in the development set. A final value of \( th = 0.47 \) over the cosine distance (in range \( 0 - 1 \)) with the nearest neighbor has been obtained.

3.3. Fusion system

Given the scores between signal speaker/face segments and the target vectors, a softmax activation have been used to normalize the scores of each modality. In order to obtain similar scores, the softmax of each modality has been applied with a different temperature \( \tau \) parameter. For speech \( \tau = 3 \) has been used and for the face modality \( \tau \) has been set to 2. For the fusion system the target/non-target distance threshold have been set to 0.03.

3.4. Results

The proposed systems have been evaluated in the RTVE2018 database for the Multimodal Speaker Diarization task of the 2018 Albayzin Challenge. The development partition is composed of one video, with a duration of around 2 hours. Enrollment data (10 still images and a short video) is provided for a total of 34 identities. The test partition is composed of three test videos, with a total duration of around 4 hours with enrollment data for 39 identities. The metric used to evaluate the systems is the Diarization Error Rate (DER), which is the sum of three different errors: Miss Speech (MISS), False Alarm (FA) and Speaker Error Rate (SER). In this challenge, the presented approaches are evaluated individually in each modality. Hence, it is needed to produce a diarization result for both speaker and face sources.

Table 1 shows the results of the presented approaches on the development partition. The first two rows correspond to the face and speaker system evaluated with their corresponding face/speaker groundtruth. Fusion system corresponds to the combination approach described in Section 2.3. Therefore, the third and fourth row of the table correspond to the fusion system evaluated with the speaker and the face groundtruth.

Speaker system shows a 41.13% DER, where the main source of error is the SER with a 31.9%. The threshold used to decide whether a segment corresponds to a target or to an unknown identity produces a low MISS but leads to a higher FA and SER. We noticed that our system failed in segments where music was included in the background and with these targets whose enrollment signal was very different to the show in terms of channel variability. Adapting the model to the RTVE corpus could have improved the rate of error caused by these factors. On the other hand, using an initial speaker segmentation on the signal instead of a sliding windows strategy could also lead to a better system performance.

For the face modality, the main source of error is the high number of missed face time (37.9%). On the other side, the FA and the SER are very low. The missed face time error could be originated from two different motives. For one side, a threshold too low could cause many false rejections of valid tracks (i.e. tracks belonging to valid enrollment identities). On the other side, this error could be originated because the face detection/tracking failed to extract valid tracks. To determine which one of these errors is predominant we have set the rejection threshold at its maximum value \( (th = 1) \), meaning that all tracks should be accepted. After doing that, we found that the missed face error was still very high (37.2%). This indicates that the errors are mainly produced by the tracking step.

The fusion system presents worst results than the the speaker and the face systems used individually. Both fusion systems evaluated with the speaker and the video groundtruths present higher MISS and SER in comparison with the monomodal systems. We have noticed that the multimodal segmentation does not improve the results because it automatically discards a lot of speaker segments and face tracks. In one hand, it discards the segments where there is no overlapping between speaker segments and face tracks. On the other hand, when more than one face appears in the video, the system automatically discards the face track of the person who is not speaking. Therefore, our assumptions would work better with the aim of looking for who is shown and speaking at the same time but not for this kind of multimodal evaluation.

4. Conclusions

We have presented two monomodal and one multimodal technologies to perform person identification in broadcast videos. A quantitative analysis has been performed on the RTVE 2018 dataset as provided in the Albayzin challenge. From the experiments it can be seen that the monomodal systems should be improved. For the speaker approach, it would be interesting to explore transfer learning methods to adapt our generic model in a smaller scenario and to include a speaker segmentation algorithm in the system. For the face modality, we plan to improve the face detection and tracking step as it has been proven that is the main source of error for the face modality. There is also a big room for improvement for the multimodal fusion system. Instead of fusing the systems from the output of the monomodal systems, an end to end multimodal system could work better if a big amount of data is available.

5. Acknowledgements

This work was supported in part by the Spanish Project DeepVoice (TEC2015-69266-P) and the project MALEGRA (TEC2016-75976-R), financed by the Spanish Ministerio de Economia, Industria y Competitividad and the European Regional Development Fund (ERDF).

6. References


Table 1: DER results on the development partition.


The GTM-UVIGO System for Audiovisual Diarization

Eduardo Ramos-Muguerza, Laura Docío-Fernández, José Luis Alba-Castro

AtlantTic Research Center, University of Vigo
eramos, ldocio, jalba@gts.uvigo.es

Abstract

This paper explains in detail the Audiovisual system deployed by the Multimedia Technologies Group (GTM) of the AtlantTic research center at the University of Vigo, for the Albayzin Multimodal Diarization Challenge (MDC) organized in the Iberspeech 2018 conference. This system is characterized by the use of state of the art face and speaker verification embeddings trained with publicly available Deep Neural Networks. Video and audio tracks are processed separately to obtain a matrix of confidence values of each time segment that are finally fused to make joint decisions on the speaker diarization result.

Index Terms: speaker recognition, face recognition, deep neural networks, image processing.

1. Introduction

In recent years, the field of pattern recognition has witnessed a shift from the extraction of handmade features to machine-learned features using complex neural network models. Biometric verification is a clear example of an application scenario where Deep Neural Networks have produced a notable increase in performance, providing transformations of space where the face and voice of users are represented in clusters that are more compact and separable than in the original sample space. This representation makes the problem of diarization and verification in multimedia content more tractable than with previous approaches [1][2][3][4].

However, facial and speaker verification models are still not perfect and make many mistakes in verifying the identity of people in natural conditions. These situations are common when analyzing audiovisual content with constant shot changes and different types of scenarios, variability in the appearance of faces (pose, expression and size), variability in the mix of voices, noise and background music. Also, the appearance of many other people who are not registered to be identified and are considered "intruders" to the system, causes many false identity assignments.

In this paper we explain the approach that the GTM research group has followed to tackle the person identification problem in audiovisual content. We have prepared a system that works separately on the video and audio tracks and makes a final fusion to fine tune the speaker diarization result. The rest of the paper is organized as follows. Section 2 explains the video processing part, including the segmentation of the video footage into different shots and the face detection, tracking, verification and post-processing at shot level. Section 3 explains the Speaker Diarization and Verification subsystem. Section 4 deals with the fusion of modalities and Section 5 gives the computational cost information. Finally, section 6 presents the conclusions and details the on-going research lines.

2. Video Processing

Television programs such as news, debates, interviews, etc., are characterized by frequent changes of shot and scene, the appearance of multiple people and the mixture of different scenes in the final configuration of frame that consumes the end user. In this way, the final audiovisual content is very different from the typical scenarios where biometric identification is used, such as restricted access, video security or mobile scenarios.

The solution we have adopted for this competition in the video processing part is based on two fundamental ideas that apply to this type of content. On the one hand, we know that a change of shot implies, in general, a change in the person who appears on the scene, although it does not always happen and it does not happen in the same way regarding the speaker. On the other hand, the people who appear in a shot remain in it as long as there is no movement of the camera or of the people themselves. This way, detection of shot changes gives an important clue for subsequent face processing.

2.1. Detection of shot changes

This subsection explains a simple approach to detect shot changes that is designed to have more false positives than false negatives. Shot changes will be used to restart face trackers because we cannot rely on tracking a face through shot changes, so losing a shot change could have a greater impact in the tracker than initializing the face tracker unnecessarily.

Detection of movement is also an important feature to have a more complete understanding of the footage, but we haven’t included in this version of the system a specific movement detection block. Instead, we have used the false positive rate of the shot detection block as an indication of movement.

The steps to detect a change of shot are the following:

1. Reduce the size of the frame to save computational load,
2. Calculate the derivatives of the image to keep the edges of the scene,
3. Divide the frame into blocks and calculate the mean of edge pixels per block,
4. Subtract the mean of the same block in the previous frame,
5. Set a threshold for considering that a block difference represents a change (threshold set with the development video footage),
6. Count the number of block changes and set a threshold defined for a change of shot (also using the development set).

This approach is very simple and fast. It also leaves a lot of room for improvement by defining areas with different shot change thresholds depending on the type of scene or type of video realization. For example, in some of the videos of the competition, the consumed scene have the frame divided into
different areas with different video content. It is quite common that the shots change at different pace in the different areas, so a solution that can make local decisions on shot change is quite useful. However, we left for future improvements of the system the local detection of shot changes. For this version we just set a permissive global threshold that allows detection of total or partial change of shot, movement and fading as a unique event.

2.2. Face processing

The face processing subsystem comprises several sequential operations that are briefly explained through the Figure 1 and in the subsections below.

2.2.1. Face Detection and Geometric Normalization

Face detection is a fundamental step in the sequential processing. We have used the detector based on Multi-Task Cascaded Convolutional neural Network [5], that jointly finds a Bounding Box for the face and five landmarking points useful to normalize the face. This face detector is quite robust to pose, expression and illumination changes. False negatives are typical in extreme poses with yaw angles beyond +/-60º and pitch angles beyond +/- 40º, that are not so uncommon in interview and debate contents. This approach also brings a bit amount of false positives in areas where textured objects with skin colors appear, like hands, arms and other not human objects.

Once a face is detected (being true detection or not), its bounding box (BB) is saved with several parameters that will allow to do tracking and assign identities during the process. An overlapping function between the current BB, and the BBs of the previous frame allows linking the BBs belonging to the same person and do backtracking when the shot has finished.

The detected face is passed to a geometric normalization that prepares the face to be plugged in in a standardized way to the face recognition block.

2.2.2. Face recognition

We have used the face recognizer based on dlib’s implementation [6] of the Microsoft ResNet DNN [1]. This DNN finds an embedded space where similar faces are grouped together and far from different faces. This network is also trained to be quite robust to pose, expression and illumination changes. In this case, the network makes quite many false identification assignments when poses are beyond +/-50º in yaw and +/- 20º in pitch. Also extreme facial expressions produce false assignments, being quite robust, though, for neutral and smiling faces (the great bulk of face images found in internet-based datasets for face recognition). However, TV contents offer more facial expressions of emotion than neutral and smiling, so false assignments are quite common also in these cases.

After a face is detected for the first time in a shot, meaning that no previous BB is linked to the current one, a candidate ID is assigned to the BB if it surpasses a distance threshold for new IDs (Th_newID) when comparing the embedded vector against all the embedded vectors of the enrollment set. The closest ID is kept for that BB. A confidence value is also assigned to that ID in that specific BB. If the BB is linked to a previous BB, the embedded vector is compared just with the cluster of enrolled vectors of the previous candidate ID. If it surpasses a second threshold (Th_previousID) then the same ID is assigned to the BB. The rationale of keeping two different thresholds and having Th_newID > Th_previousID is that assigning an ID to a new detection should be more restrictive than assigning it to a previous one located in an overlapped area. It is important to highlight that a shot change can make a face with different ID to appear in the same position that another face in the previous frame before the shot. So, a change of shot restarts the tracking process.

2.2.3. Face clusters augmentation

The unreliable robustness of the face recognizer with extreme poses and expressions lead us to define a strategy to cope with potential erroneous ID assignments.

A previous study of the capacity of dlib’s facial recognition network, advised us to use a restrictive Th_newID, so comparison of faces with extreme poses between the current embedding and the embeddings in the enrollment dataset will not surpass it. This way, it is less likely that a wrong ID assignment is propagated through a tracked BB. Only when the head moves to a more frontal pose, or a similar extreme pose is stored in the enrollment dataset, the ID is assigned to the BB and propagated. But, what happens if the face in the shot is always in an extreme pose? We need a way to enrich the enrollment dataset with new samples of a specific ID that appear in the content but are different enough from the enrollment dataset. Given that the Th_previousID is more relaxed, during a shot where a previous match has been assigned, several different face samples with almost any pose and expression can enrich the enrollment dataset. The criterion to enrich the dataset is the increase of variance of the ID face cluster. Then, the enriched dataset is ready to use in the next frame.

2.2.4. Face ID backtracking

Once a shot change is detected, an online post-processing is run to reassign IDs or even delete potentially wrong assignments in the past shot. This block is based on heuristic rules defined after observing the typical behavior of the previous processing blocks in the development scenarios. If the detector and recognizer were ideal, a shot without rapid movements or with slow camera movement should contain just a number of BB tracks that matches the number of persons in the shot; the first BB should appear in the first shot frame and the last BB of its BB track should appear in the last frame of the shot. But things are not perfect, so the rationale of the backtracking post-processing is based on the next observations:

- False negatives of the detector break the paths of the BBs, so there will be more paths than in the ideal case.
- False initial matchings in every new track will yield tracks with incorrect IDs.
- The number of persons in the shot can be roughly estimated by the average number of detections. This assumption is broken when extreme poses appear during a long period, producing an underestimated number of persons in the shot.
- Every ID matched in the shot can appear in several broken paths. The accumulated confidence of that ID gives a rough approximation of its probability of being in that shot.

Using these observations we defined a rule to keep tracks of a number of persons just above the estimated average and with the IDs with largest accumulated confidence. Those IDs are finally assigned to the time intervals within and across shots and written down in an rttm file.
3. Speaker Diarization and Recognition

The developed strategy for speaker diarization and verification uses a DNN trained to discriminate between speakers, and which maps variable-length utterances or speech segments to fixed-dimensional embeddings that are also called x-vectors [2].

A pretrained deep neural network downloaded from http://kaldi-asr.org/models.html was used. The network was implemented using the nnet3 neural network library in the Kaldi Speech Recognition Toolkit [7] and trained on augmented VoxCeleb 1 and VoxCeleb 2 data [8].

Figure 2 represents the block diagram of the speaker diarization and recognition subsystem.

3.1. Speaker enrollment

The audio signal provided for each person in the enrollment set is used to obtain DNN speech-based embeddings. A sliding window of at least 10 seconds with a half a second hop is used. Then, these embeddings are clustered using the Chinese Whispers algorithm [9]. The threshold of the clustering algorithm is adjusted so that the clusters are pure and at least as many as the number of identities in the enrollment set. In this way an enrolled person can be represented by one or more clusters.

3.2. Off-line Speaker Diarization

First, the audio signal is divided into 3 second segments with a half a second hop. DNN short-term audio embeddings were extracted for each of these segment, clustered using the Chinese Whispers algorithm and their timestamps kept. From the clustering result we obtain an audio segmentation. Next, each of these segments, of arbitrary duration, are processed in order to extract one or more long-term audio embeddings using the same DNN. To do this, a sliding window of at least 10 seconds with a half a second hop is used. Then, these embeddings are clustered using again the Chinese Whispers algorithm, using a threshold that minimizes the diarization error.

Figure 2: Block diagram of the speaker diarization and recognition subsystem.

Figure 1: Flow diagram of face processing.
3.3. On-line Identity Assignment

The clusters obtained in the previous step need to be assigned to the enrollment identities. Keeping the timestamps of each embedding in the clustering process, allows to design an online ID assignment approach. Temporal segments are defined as consecutive timestamps with embeddings associated to the same cluster. The ID assigned to a time segment is the enrollment ID of the best-matching enrollment cluster, as far as this distance is less than a threshold. This threshold is defined after observing the typical behavior of the system in the development scenarios. A confidence value for that ID in that specific temporal segment is stored to be used jointly with the face-based confidence value in the fusion process.

4. Fusion

Once a decision was made for both modalities, a multimodal fusion approach was implemented in order to correct potentially wrong speech-based ID assignments. Given a temporal segment that has been assigned a speaker identity ID1, if a high-confidence single face identity ID2 has been detected in more than 60% of the video frames in that speaker ID1 segment, the speaker ID1 is changed to the face identity ID2.

This rule doesn’t apply if ID1 and ID2 have different gender (as given by the enrollment name).

Much more elaborated fusion rules can be applied at this stage, but only this one was tested for the competition.

The results over the Development video provided by the organizers of the competition are presented in Table 1.

<table>
<thead>
<tr>
<th>Modality</th>
<th>DER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face</td>
<td>36.20%</td>
</tr>
<tr>
<td>Speaker</td>
<td>14.25%</td>
</tr>
<tr>
<td>Speaker Fusion</td>
<td>7.51%</td>
</tr>
<tr>
<td>Average DER</td>
<td>21.85%</td>
</tr>
</tbody>
</table>

5. Computational Cost

The computational cost of the proposed audiovisual diarization system was measured in terms of the real-time factor (RT). This measure represents the amount of time needed to process one second of audiovisual content: \( x_{RT} = P/I \), where \( I \) is the duration of the processed video and \( P \) is the time required for processing it. The whole development video was processed to compute the RT, thus taking into account many different audiovisual situations. The duration of this video is \( I = 7410 \) s, and the time needed to process it was \( P = 33457 \) s, leading to \( RT = 4.51 \). These computation time was obtained by running this experiment on an Intel(R) Core(TM) i5 CPU 670@3.47 GHz with 12 GB RAM. Even though the process is running more than 4 times slower than real-time, the code is not optimized at all (some parts are coded in Matlab) and the machine is just using 1 CPU and no GPU. We are working to speed up the process and expect to have it running in real-time in the next months.

6. Conclusions and future work

We have presented the GTM-UVIGO System deployed for the Albayzin Multimodal Diarization Competition at Iberspeech 2018. The system uses state of the art DNN algorithms for face detection and verification and also for speaker diarization and verification. The application scenario is studied to implement ad-hoc post-processing strategies to fine-tune the ID assignments made by the video and audio parts. Specifically, the information on shot changes are exploited to avoid tracking faces across shots. Confidence matrices are used in a fusion strategy that allows changing pre-assigned speaker identities.

This framework leaves a lot of room for improvement in each of the fundamental processing stages and also in the ad-hoc rules for fine-tuning. One of the main future lines consist of increasing the robustness of face matchings for extreme poses and expressions and the robustness of speaker ID assignments in overlapped speech. From a video processing point of view, a better characterization of the display montage would allow the application of post-processing rules less prone to errors. Finally, speeding up some DNN critical parts by using GPUs and efficiently coding some other parts would allow real-time processing of the system.

Acknowledgements

This work has received financial support from the Spanish Ministerio de Economia y Competitividad through project ‘TraceThem’ TEC2015-65345-P, from the Xunta de Galicia (Agrupación Estratéxica Consolidada de Galicia accreditation 2016-2019) and the European Union (European Regional Development Fund ERDF).

References

The SRI International STAR-LAB System Description for IberSPEECH-RTVE 2018 Speaker Diarization Challenge

Diego Castan, Mitchell McLaren, Mahesh Kumar Nandwana

Speech Technology and Research Laboratory, SRI International, California, USA
{diego.castan, mitchell.mclaren, maheshkumar.nandwana}@sri.com

Abstract
This document describes the submissions of STAR-LAB (the Speech Technology and Research Laboratory at SRI International) to the open-set condition of the IberSPEECH-RTVE 2018 Speaker Diarization Challenge. The core components of the submissions included noise-robust speech activity detection, speaker embeddings for initializing diarization with domain adaptation, and Variational Bayes (VB) diarization using a DNN bottleneck i-vector subspaces.

1. Introduction
SRI International has long focused on the task of speaker recognition, but has only recently branched into the field of speaker diarization. For the STAR-LAB submissions, we leveraged the embeddings and diarization systems that we recently developed for the NIST 2018 Speaker Recognition Evaluation [1]. In addition, we attempt to leverage recent work in speech activity detection and in speaker embeddings for speaker recognition [2, 3, 4], the well-known Variational Bayes (VB) approach to diarization [5], and the use of a DNN bottleneck based i-vector subspaces internal to the VB process. We describe the three systems submitted for the open-set condition based on a hybrid embeddings-VB approach with different parameters for the speech activity detection system.

2. System Training Data
System training data included 234,288 signals from 14,630 speakers. This data was compiled from NIST SRE 2004-2008, NIST SRE 2012, Mixer6, Voxceleb1, and Voxceleb2 (train set) data. Voxceleb1 data had 60 speakers removed that overlapped with Speakers in the Wild (SITW), according to the publicly available list1.

Augmentation of data was applied using four categories of degradations as in [4], including music, and noise at 5 dB signal-to-noise ratio, compression, and low levels of reverb. We used 412 noises of at least 15 seconds in length compiled from both freesound.org and the MUSAN corpus. Music degradations were sourced from 645 files from MUSAN, and 99 instrumental pieces purchased from Amazon music. For reverberation, examples were collected from 47 real impulse responses available on echothief.com, and 400 low-level reverb signals sourced from MUSAN. The random selection of reverb signals gave almost 10x weight to the echothief.com examples in order to balance data sources. Compression was applied using 32 different codec-bitrate combinations with open source tools such as FFmpeg, codec2, Speex, GSM, and opus trans-coding packages. In addition to these augmentations, we down-sampled any 16k or higher data (74,447 files) to 8k before up-sampling to 16k, which we have found to allow the embeddings DNN to generalize across the 8-16 kHz bandwidth range and better accommodate processing of telephone signals.

We augmented the raw speaker embeddings training data (counting the 16k re-sampled to 8k as raw data) to produce 2 copies per file per degradation type (random selection of specific degradation) such that the data available for training was 9-fold the original amount. In total, this was 2,778,615 files for training the speaker embedding DNNs. For PLDA training used for clustering embeddings [6], the same degraded data (excluding the 8k simulated data) was subsampled by a random factor of 6 in order to make PLDA training data manageable and resulted in 343,535 files from 11,461 speakers. Finally, the databases used to train the UBM and the total variability subspaces were Fisher, Switchboard and AMI PRISM (NIST SRE04-08).

3. The STAR-LAB System Submissions
We start with a general overview of the submissions prior to breaking down into details of each module used in the system. Figure 1 shows a block diagram of the different parts of the system. All our submissions use embeddings clustering as seed to VB diarization. The differences between the primary and the contrastive submissions are the thresholds applied for the SAD decisions as detailed in Section 3.2

3.1. Acoustic Features and Bottleneck i-vector extraction
To train the DNN we used Power-Normalized Cepstral Coefficients (PNCC) [7] with 30 dimensions extracted from a bandwidth of 100-7600 Hz using 40 filters. Root compression of 1/15 was applied. All audio was sampled (up or down sampled where needed) to 16 kHz prior to processing. The features were first processed with mean and variance normalization over a sliding window of 3 seconds prior to having SAD applied.

We used Mel Frequency Cepstral Coefficients for speech activity detection and as input to a DNN to extract 80-dimensional bottleneck (BN) features. This BN DNN was trained for the task of automatic speech recognition. The DNN was trained to predict 1933 English tied tri-phone states (senones). MFCCs were used for input to the DNN after transforming them with a pcaDCT transform [8] trained on Fisher data, and restricting the output dimension to 90. The DNN consisted of 5 hidden layers of 600 nodes, except the last hidden layer which was 80 nodes and formed the bottleneck layer from with activations were extracted as features.

BN features were used to train an i-vector extractor [9] consisting of a 2048-component universal background model (UBM) with diagonal covariance and a subspace of rank 400.

---

1http://www.openslr.org/resources/49/voxceleb1_sitw_overlap.txt
Figure 1: Flow diagram of components used in the STAR-LAB team submissions to the IberSPEECH-RTVE 2018 Speaker Diarization Challenge.

3.2. Speech Activity Detection

We used a DNN-based Speech Activity Detection (SAD) model leveraging short-term normalization. The SAD model was trained on clean telephone and microphone data from a random selection of files from the provided Mixer datasets (2004-2008), Fisher, Switchboard, Mixer6, SRE’18 unlabeled and SRE’16 unlabeled data. A 5 minute DTMF tone (acquired from YouTube), and a selection of noise and music samples with and without speech added were added to the pool of data. In all, 11,668 files were used to train the SAD model.

The system uses 19-dimensional MFCC features, which excluded C0 and used 24 filters over a bandwidth of 200-3300 Hz. These features were mean and variance normalized using a sliding window of 3 seconds, and concatenated over a window of 31 frames. The resulting 620-dimensional feature vector formed the input to a DNN which consisted of two hidden layers of sizes 500 and 100. The output layer of the DNN consisted of two nodes trained to predict the posteriors for the speech and non-speech classes. These posteriors are converted into likelihood ratios using Bayes rule (assuming a prior of 0.5), and thresholded at a value of -1.5, -2.0 and -3.0 for the primary and both contrastive systems, respectively. A padding of 0.5 seconds was applied over the final segmentation to smooth the transitions between speech/non-speech.

We applied cross-talk removal on all interview data from the NIST SRE corpora to suppress the interviewer speech that bled through to the target speaker channel. This was especially important for distant microphone channels in which each speaker had similar energy. Cross-talk removal involved using the SAD Log-Likelihood Ratios (LLRs) from the target microphone as well as the close-talking interviewer microphone, and removing any detected speech from the target channel that was detected in the interviewer channel with more than 3.5 in LLR value of the target channel.

For all augmented system training data, the SAD alignments from the raw audio were used rather than running SAD on the degraded signals directly, as done in [4].

3.3. Embeddings VB Initialization

Recent work in [2, 3] has shown significant advances in the related field of speaker recognition by replacing the well-known i-vector extraction process with speaker embeddings extracted from a DNN trained to directly discriminate speakers. We decided to apply our findings on what makes a good speaker embeddings extractor [4] to the task of speaker clustering.

We used a multi-bandwidth speaker embedding DNN in our submission. Speaker embedding DNNs were trained following the protocol of [4]. Specifically, Kaldi was used to generate examples for training the DNN with a duration ranging between 2.0 and 3.5 seconds of speech. DNNs were trained using Tensorflow over 6 epochs using a mini batch size of 128 examples, and dropout probability linearly increasing to 10% then back to 0% in the final 2 epochs. The embeddings network starts with five frame-level hidden layers, all using rectified linear unit (ReLU) activation and batch normalization. The first three layers incrementally add time context with stacking of [-2,-1,0,1,2], [-2,0,2], and [-3,0,3] instances of the input frame. A statistics pooling layer then stacks the mean and standard deviation of the frames per audio segment, resulting in a 3000 dimensional segment-level representation. The final two hidden layers of 512 nodes operate at the segment-level and use ReLU activation and batch normalization prior to the output layer, which targets speaker labels for each audio segment using log softmax as the output. The embeddings are extracted from the first segment-level hidden layer of 512 nodes. This system used PLDA classification for clustering after applying an LDA dimensionality reduction. We applied length and mean normalization to embeddings prior to use in PLDA. As a simple method of domain adaptation, we mean normalized the chunked embeddings from an audio file using the mean of all chunks.

The embeddings VB initialization process was performed as follows. The audio was first segmented into 1.5 second segments with 0.2 second shift. Following a similar strategy to VB diarization, we initialized a speaker cluster posterior matrix, q, to 13 speakers. The number of speakers was selected from previous experiments over the development data. We calculated for each speaker cluster, a weighted-average embedding based on q and the 1.5s embeddings segments. These per-cluster embeddings were compared using PLDA against each individual embedding segment. We scaled the likelihood ratios (LLRs) that resulted from PLDA by 0.05 and performed Viterbi decoding of the LLRs to result in a new q and speaker priors. This process was iterated 10 times before using the result q and speaker priors in the subsequent VB diarization based on BN+MFCC features.

3.4. Variational Bayes diarization

Our diarization approach was based on the work of [10]. This approach uses an i-vector subspace to produce a frame-level diarization output. Our i-vector subspace was trained using concatenated BN and MFCC features [11], resulting in a feature with 140 dimensions. With VB diarization, we have used a left-to-right HMM structure of three states per speaker in order to smooth the transitions between speakers that was proposed in [12].

The initialization of the VB diarization approach is done with the speaker posteriors estimated from the speaker embeddings initialization. We performed a maximum of 20 iterations of VB diarization.

4. Results

This section compares the development performance of the STAR-LAB submissions.

Table 1 shows the system performances for the primary and the two contrastive systems in the development set.
Table 1: Development results for each system submission

<table>
<thead>
<tr>
<th>System Name</th>
<th>Miss Sp</th>
<th>FA Sp</th>
<th>SpkErr</th>
<th>DER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>2.1%</td>
<td>1.9%</td>
<td>13.4%</td>
<td>17.38%</td>
</tr>
<tr>
<td>Contrastive 1</td>
<td>2.0%</td>
<td>2.3%</td>
<td>13.4%</td>
<td>17.81%</td>
</tr>
<tr>
<td>Contrastive 2</td>
<td>2.0%</td>
<td>2.9%</td>
<td>12.2%</td>
<td>17.08%</td>
</tr>
</tbody>
</table>

Table 2: Computational requirements of STAR-LAB submissions from based on RT factor (higher than 1.0 is slower than real time) and maximum resident memory needed to diarize 10-minutes of development file millennium-20170522.

<table>
<thead>
<tr>
<th>System</th>
<th>x RT</th>
<th>Max. Res. RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>1.19</td>
<td>3.6G</td>
</tr>
<tr>
<td>Contrastive 1</td>
<td>1.52</td>
<td>3.6G</td>
</tr>
<tr>
<td>Contrastive 2</td>
<td>1.28</td>
<td>3.6G</td>
</tr>
</tbody>
</table>

5. Computation

We benchmarked the computational requirements of the STAR-LAB system on a single core. The machine was an Intel Xeon E5630 Processor operating at 2.53GHz. The approximate processing speed and resource requirements are listed in Table 2. These calculations are based on total CPU time divided by the total duration of the audio.

6. Acknowledgments

We'd like to thank Lukas Burget and the BUT team for their python implementation of Variational Bayes diarization which was leveraged in this work [10].

7. References


ODESSA at Albayzin Speaker Diarization Challenge 2018

Jose Patino¹, Héctor Delgado¹, Ruiqing Yin², Hervé Bredin², Claude Barras³ and Nicholas Evans¹

¹EURECOM, Sophia Antipolis, France
²LIMSI, CNRS, Université Paris-Saclay, Orsay, France
³LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, Orsay, France

1 firstname.lastname@eurecom.fr, 2³firstname.lastname@limsi.fr

Abstract

This paper describes the ODESSA submissions to the Albayzin Speaker Diarization Challenge 2018. The challenge addresses the diarization of TV shows. This work explores three different techniques to represent speech segments, namely binary key, vector and triplet-loss based embeddings. We place the short-time Fourier transform by an infinite-impulse response on a scenario where training data is limited, the training of robust neural-embedding extractors is considerably more challenging. However, when training data is plentiful (open-set condition), neural embeddings provide more robust segmentations, giving speaker representations which lead to better diarization performance. The paper also reports our efforts to improve speaker diarization performance through system combination. For systems with a common temporal resolution, fusion is per-formed at segment level during clustering. When the systems under fusion produce segmentations with an arbitrary resolution, they are combined at diarization hypothesis level. Both approaches to fusion are shown to improve diarization performance.

Index Terms: speaker diarization, diarization fusion, neural embeddings, binary key

1. Introduction

Albayzin evaluations cover a range of speech processing related tasks that include search on speech, audio segmentation, speech-to-text transcription, or speaker diarization. The latter motivates the work reported on this paper. Speaker diarization is the task of processing an audio stream into speaker homogeneous clusters. Traditionally considered as an enabling technology, a number of potential applications can benefit from speaker diarization as a pre-processing step, such as automatic speech recognition [1], speaker recognition and identification [2], or spoken document retrieval. The increasing maturity of these technologies calls for continuous improvement in speaker diarization that has accordingly been the objective of numerous evaluations and campaigns, be it in the context of the NIST RT evaluations [3], the more recent multi-domain DIHARD challenge [4], or in the well-established Albayzin Speaker Diarization evaluations [5]. The current edition of the Albayzin Speaker Diarization Challenge [6] includes audio content from the recently released RTVE2018 database [7], composed of TV shows from a range of topics broadcast on the Spanish TV public network. Further details can be found in [6, 7].

This work has been produced in the context of the ODESSA project, which is focused on improving speaker diarization performance by leveraging recent developments in the task of text-independent speaker recognition. Efforts were made by the different members of the consortium to improve the reliability of their own speaker diarization systems. In doing so, different speaker modelling techniques were used, on both the closed- and open-set conditions of the evaluation. For the closed-set condition, binary key speaker modelling [8] offers a training-free option that has produced competitive performance in previous editions of the challenge [9]. Triplet-loss neural embeddings [10] trained on the provided data were also explored. Experiments in the open-set condition include embeddings in the form of state-of-the-art text-independent speaker recognition x-vector [11]. Different clustering techniques were also explored across the training conditions and speaker modelling techniques.

Finally, and motivated by the access to significantly different approaches to speaker modelling and diarization that could potentially offer complementary solutions, the main contribution of this work lies in the exploration of different fusion techniques for speaker diarization. Whereas fusion, for example at score level, is often applied as a mean of increasing robustness in closely related tasks such as speaker recognition, the problem of merging clustering solutions for speaker diarization scenarios remains challenging. Diarization hypothesis level fusion is applied following a label merging approach [12]. A segment-level fusion technique similar to that employed in [13] is also tested.

The remainder of this paper is structured as follows. Section 2 details the processing blocks which compose the different diarization solutions. Section 3 describes the two fusion approaches. Sections 4 and 5 report the experimental setup, and submitted systems with results, respectively. Finally, Sections 6 and 7 provide discussions and conclusions.

2. Processing modules

This section reviews the different processing modules composing our diarization systems. These include feature extraction, speech activity detection, segmentation, segment and cluster representation, clustering and re-segmentation. As shown in Figure 1, one or more techniques are proposed for each processing module.

2.1. Feature extraction

Two different acoustic frontends were used. They include (i) a standard Mel-frequency cepstral coefficients (MFCC) [14] frontend and (ii) an infinite impulse response - constant Q, Mel-frequency cepstral coefficients (ICMC) [15] frontend. The latter has been applied successfully to tasks including speaker recognition, utterance verification [15] and speaker diarization [9, 16]. These features are similar to MFCC, but they replace the short-time Fourier transform by an infinite-impulse re-
2.2. Speech activity detection and segmentation

All submissions share a common speech activity detection (SAD) module [18], where SAD is modelled as a supervised binary classification task (speech vs. non-speech), and addressed as a frame-wise sequence labelling task using a bi-directional long short-term memory (LSTM) network operating on MFCC features. As for segmentation, two systems were explored: (i) a straightforward uniform segmentation which splits speech content into 1 second segments and (ii) segmentation via the detection of speaker change points. The speaker change detection (SCD) module is that proposed in [19]. Similarly to the SAD module, SCD is also modelled here as a supervised binary sequence labelling task (change vs. non-change).

2.3. Segment/cluster representation

Binary key. This technique was initially proposed for speaker recognition [8, 20] and applied to speaker diarization [21, 9, 16]. It represents speech segments as low-dimensional, speaker-discriminative binary or integer vectors, which can be clustered using some sort of similarity measure. The core model to perform this mapping is a binary key background model (KBM) which is trained in the test segment before diarization. The KBM is actually a collection of diagonal-covariance Gaussian models selected from a pool of Gaussians learned on a sliding window over the test data. The window rate is adjusted dynamically to assure a minimum number of Gaussians. Then, a selection process is performed to keep a percentage of the Gaussians in the pool to ensure sufficient coverage of all the speakers in the test audio stream. The KBM is then used to binarize an input sequence of acoustic features, which are then accumulated to obtain a cumulative vector, which is the final representation. Refer to [21] for more details.

Triplet-loss neural embedding. The embedding architecture used is the one introduced in [10] and further improved in [22]. In the embedding space, using the triplet loss paradigm, two sequences \( x_i \) and \( x_j \) of the same speaker (resp. two different speakers) are expected to be close to (resp. far from) each other according to their angular distance.

x-vector. This method [11] uses a deep neural network (DNN) which maps variable length utterances to fixed-dimensional embeddings. The network consists of three main blocks. The first is a set of layers which implements a time-delay neural network (TDNN) [23] which operates at the frame level. The second is a statistics pooling layer that collects statistics (mean and variance) at the utterance level. Finally a number of fully connected layers are followed by the output layer with as many outputs as speakers in the training data. Neurons of all layers use ReLu activations except the output layer neurons which use soft-max. The network is trained to discriminate between speakers in the training set. Once trained, the network is used to extract utterance-level embeddings for utterances from unseen speakers. The embedding is just the output of one of the fully connected layers after the statistics pooling layer.

2.4. Clustering

Agglomerative hierarchical clustering. The AHC clustering uses a bottom-up agglomerative clustering algorithm as follows. First, and assuming that the input audio stream is represented as a matrix of segment-level embeddings, a number of clusters \( M_{init} \) are initialised by a uniform splitting of the segment-level embedding matrix. Cluster embeddings are estimated as the mean segments embeddings. An iterative process including: (i) segment-to-cluster assignment, (ii) closest cluster pair merging and (iii) cluster embedding re-estimation by averaging embeddings of cluster members is then applied. All comparisons are performed using the cosine similarity between embeddings. The clustering solutions generated after (i) are stored at every iteration. The output solution is selected by finding a trade-off between the number of clusters and the within-class sum of squares (WCSS) among all solutions. This is accomplished through an elbow criterion, as described in [21].

Affinity propagation. As proposed in [24], an affinity propagation (AP) algorithm [25] is our second clustering method. In contrast to other approaches, AP does not require a prior choice of the number of clusters contrary to other clustering methods. All speech segments are potential cluster centres (exemplars). Taking as input the pair-wise similarities between all pairs of speech segments, AP will select the exemplars and associate all other speech segments to an exemplar. In our case, the similarity between the \( i^{th} \) and \( j^{th} \) speech segments is the negative angular distance between their embeddings.

2.5. Re-segmentation

A resegmentation process is performed to refine time boundaries of the segments generated in the clustering step. It uses Gaussian mixture models (GMM) to model the clusters, and maximum likelihood scoring at feature level. Since the log-likelihoods at frame level are noisy, an average smoothing within a sliding window is applied to the log-likelihood curves obtained with each cluster GMM. Then, each frame is assigned to the cluster which provides the highest smoothed log-likelihood.

3. System fusion

Two approaches to fusion were explored. The first operates at the similarity matrix level suited to combine speaker diarization systems that are aligned at the segment level. The second operates at the hypothesis level and can be applied to systems with...
Figure 2: Illustration of the segment-to-cluster similarity matrix fusion.

arbitrary segment resolutions.

**Fusion at similarity matrix level.** Systems sharing the same segmentation can be combined at the similarity level. In [13] fusion is performed by the weighted sum of the similarity matrices of two segment-aligned systems before a linkage agglomerative clustering. This approach was adapted to our AHC algorithm by combining segment-to-cluster and cluster-to-cluster similarity matrices at every iteration. Similarities are also combined in the WCSS computation for the best clustering selection procedure. In this way, the full process takes into account the influence of the systems being fused. An example of combination of two $M$-cluster to $N$-segments similarity matrices using weights $\alpha$ and $1 - \alpha$ is depicted in Figure 2.

**Fusion at hypothesis level.** The combination of systems with totally diarization pipelines is generally only possible at hypothesis level. In this work we explored hypothesis level combination using the approach described in [12]. Given a set of diarization hypotheses, every frame-level decision can be merged to assign a new frame-level cluster label which is the concatenation of all labels of the individual hypotheses. An example of this strategy is illustrated in Figure 3. This process will result in a large set of potential speaker clusters. Clusters shorter than 15 seconds are excluded and a final resegmentation is applied on the merged diarization to obtain the final diarization hypothesis.

### 4. Experimental setup

This section gives details of the training data and the configuration of the different modules.

#### 4.1. Training data

For the closed-set condition, the 3/24 channel database of around 87 hours TV broadcast programmes in Catalan language provided by the organisers was used. For the open-set condition, two popular datasets were used:

**SRE-data.** It includes several datasets released over the years in the context of the NIST speaker and recognition evaluations (SRE), namely SRE 2004, 2005, 2006, 2008 and 2010, Switchboard, and Mixer 6. This dataset contains mostly telephone speech sampled at 8 kHz.

**VoxCeleb.** The VoxCeleb1 dataset [26] consists of videos containing more than 100,000 utterances for 1,251 celebrities extracted from YouTube videos. The speakers represent a wide range of different ethnicities, accents, professions and ages, and a large range of acoustic environments. This dataset is sampled at 16 kHz.

#### 4.2. System configuration

**Feature extraction.** MFCCs are extracted with different numbers of coefficients depending on the subsequent segment representation: 23 static coefficients for x-vector, and 19 plus energy augmented with their first and second derivatives for triplet-loss embeddings. The binary key system uses 19 static ICMC features. Finally, the re-segmentation stage uses 19 static MFCC features.

**Segment representation.** For BK, the cumulative vector dimension is set to $p = 40\%$ of the size of the initial pool of Gaussian components in the KBM, leading to different representation dimensions which depend on the length of the test audio file. Gaussians are learned on a sliding window of 2 seconds to conform a pool with a minimum size set to 1024. The x-vector system uses the configuration employed in the Kaldi recipe for the SRE 2016 task\(^2\). Data augmentation by means of additive and convolutive noise is performed for training. The dimension of the embeddings is 512, which was later reduced to 170 using LDA. For triplet-loss embeddings, and because of the lack of global identities in the Albayzin dataset, triplets are only sampled from intra-files for the closed-set condition. Thanks to the given speaker names in Voxceleb, triplets are also sampled from inter-files for the open-set condition.

**Clustering.** AHC is initialised to a number $N_{\text{init}}$ of cluster higher than the number of expected clusters in the test sessions. We set $N_{\text{init}} = 30$. The parameters of AF clustering such as preference and damping factor are tuned on the development set with the chocolate toolkit\(^3\).

**Resegmentation.** It is performed with GMMs with 128 diagonal-covariance components. Likelihoods are smoothed by a sliding window of 1s.

#### 4.3. Evaluation

Performance is assessed and optimised using the diarization error rate (DER), a standard metric for this task. It is defined as $\text{DER} = \text{ER}_{\text{spkr}} + \text{ER}_{\text{FA}} + \text{ER}_{\text{MS}}$, where $\text{ER}_{\text{FA}}$ and $\text{ER}_{\text{MS}}$ are factors mainly associated to the SAD module, namely the false alarm and miss speech error rates. $\text{ER}_{\text{MS}}$ is also incremented by the percentage of overlapping speech present in the audio sessions that is insufficiently assigned to a single speaker. Our systems do not contain any purpose-specific module to detect such segments. Finally, $\text{ER}_{\text{spkr}}$ is the error related to speech segments that are associated to the wrong speaker identities. It is

---

\(^2\)https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2

\(^3\)https://chocolate.readthedocs.io/
Table 1: Summary of ODESSA Primary (P) and contrastive (C1/C2) submissions for the closed- and open-set (denoted by c and o subscript, respectively) conditions, including feature extraction, segmentation and training data used, segment representation, clustering and fusion. Performance (DER, %) is shown in the last column.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Sys.</th>
<th>Features</th>
<th>Segmentation</th>
<th>Segment rep. / train data</th>
<th>Clustering</th>
<th>Fusion</th>
<th>DER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closed</td>
<td>P_c</td>
<td>ICMC</td>
<td>1-second</td>
<td>BK / -</td>
<td>-</td>
<td>C1_c, C2_c, Hyp-level</td>
<td>10.17</td>
</tr>
<tr>
<td></td>
<td>C1_c</td>
<td>MFCC</td>
<td>BiLSTM</td>
<td>EMB / 3/24 data</td>
<td>AHC</td>
<td>-</td>
<td>12.33</td>
</tr>
<tr>
<td></td>
<td>C2_c</td>
<td></td>
<td></td>
<td></td>
<td>AP</td>
<td></td>
<td>14.10</td>
</tr>
<tr>
<td>Open</td>
<td>P_o</td>
<td>IC</td>
<td>1-second</td>
<td></td>
<td>AHC</td>
<td>C1_o, C1_c, C2_o, Sim-level</td>
<td>7.21</td>
</tr>
<tr>
<td></td>
<td>C1_o</td>
<td>MFCC</td>
<td>x-vector / Voxceleb</td>
<td>AHC</td>
<td>-</td>
<td>9.29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C2_o</td>
<td></td>
<td></td>
<td></td>
<td>AP</td>
<td></td>
<td>11.46</td>
</tr>
</tbody>
</table>

6. Discussion

Open- vs. closed-set conditions. As shown in Table 1, open-set systems outperform closed-set ones on the development set. This is somehow expected since neural approaches usually benefit from large amounts of data. Apart from differences in the amount of training data, the absence of global speaker IDs on the closed-set training data are also likely to influence performance. Our attempt to train an x-vector extractor for the closed-set condition was unfruitful, and performance of the triplet-loss embedding extractor was significantly worse than when using external training data (system C2_o vs. C2_c). In this situation, the simpler, training-free binary key approach turned out to be the single best performing system (C1_c), showing the potential of such techniques when training data is sparse or unavailable.

Segmentation. It is not clear if a dedicated module to speaker turn detection brings significant benefits compared to simple, straightforward uniform segmentation of the input stream. A volume of work reports the relative success of both explicit approaches to SCD [27, 28, 24] and uniform segmentation approaches [13, 29, 16]. In this work we found that some of the speaker representations work better with either one of the two strategies, with the uniform approach being better suited to BK and x-vector, and SCD to the triplet-loss embeddings.

System combination. In the closed-set condition, and because of the segmentation mismatch of our best performing single systems, they could not be combined at the similarity matrix level. Hence, the most obvious combination was at hypothesis level. The label combination procedure followed by a re-segmentation resulted in a lower DER than those of the two individual systems. In the open-set condition three subsystems used the same segmentation, and hence they could be fused at the similarity matrix level. This enabled the clustering to be performed jointly by considering the contributions of the three subsystems in parallel along the complete diarization process.

7. Conclusions

This paper reports the participation of the ODESSA team to the Albayzin Speaker Diarization Challenge 2018. As a consortium, our main interest was on the combination of multiple approaches to diarization. We assessed the effectiveness of two fusion strategies, namely at similarity matrix level, and at diarization hypothesis level, which allowed the combination of segment-time-aligned and arbitrary time-aligned diarization algorithms, respectively. We also found the use of appropriate and abundant training data was critical to the learning of robust embeddings, while training-free approaches are demonstrated to be adequate in the absence of suitable training data. For future work, and together with other classical challenges such as the problem of overlapping speakers, the impact of speaker change detection should be further investigated.
8. References


EML Submission to Albayzin 2018 Speaker Diarization Challenge

Omid Ghahabi, Volker Fischer

EML European Media Laboratory GmbH, Berliner Straße 45, 69120 Heidelberg, Germany
omid.ghahabi@eml.org; volker.fischer@eml.org

Abstract

Speaker diarization, who is speaking when, is one of the most challenging tasks in speaker recognition, as usually no prior information is available about the identity and the number of the speakers in an audio recording. The task will be more challenging when there is some noise or music on the background and the speakers are changed more frequently. This usually happens in broadcast news conversations. In this paper, we use the EML speaker diarization system as a participation to the recent Albayzin Evaluation challenge. The EML system uses a real-time robust algorithm to make decision about the identity of the speakers approximately every 2 sec. The experimental results on about 16 hours of the developing data provided in the challenge show a reasonable accuracy of the system with a very low computational cost.

Index Terms: Speaker Diarization, Albayzin Evaluation, Online.

1. Introduction

Speaker diarization is the process of identifying who is speaking when during an audio recording. Compared to other speaker recognition tasks, speaker diarization is usually more difficult as there is no prior knowledge about the number and the identity of the speakers speaking in the audio recording. The task has received more attention by the speech community with increasing the number of broadcast, meeting, and call center recordings collected every year.

Albayzin evaluations are organized every two years and try to challenge the current unsolved problems in the speech processing area and are supported by the Spanish Thematic Network on Speech Technology (RTTH). One of the challenges this year has been speaker diarization of broadcast news recordings. The task itself is not new but it gets really challenging when the data is noisy, there is a background music when the speakers talk, and when speakers speak at the same time. Unlike the last Albayzin speaker diarization evaluation, the speech/nonspeech labels of the audio recordings are not available which makes the task more difficult.

In addition to the data of the previous evaluations, the new data is provided by the public Spanish National Television (RTVE) this year. The database comprises different programs broadcast by RTVE from 2015 to 2018. The programs cover a great variety of scenarios from studio to live broadcast, from read speech to spontaneous speech, different Spanish accents, including Latin-American accents and a great variety of contents [1].

Like in previous evaluations, two participating conditions are possible, close-set and open-set. In the close-set condition, only the data provided in the challenge is allowed to be used for the system development while for the open-set condition, the participating sites can use also other publicly available datasets in addition to the data provided in the challenge. We have participated only in the close-set condition in this paper.

We have used the EML Online speaker diarization system in this challenge. The system decides about the identity of the speakers at once every approximately 2 sec without any knowledge about the upcoming segments. The system uses the robust Voice Activity Detection (VAD) proposed in [2]. The algorithm used in the system is robust with a very low computational cost. The whole speaker diarization process including feature extraction is performed in an approximately 0.01 × RT.

The rest of the paper is organized as follows. Section 2 describes clearly the databases used to build the system. Section 3 describes briefly the algorithm and every part of the EML speaker diarization system. Section 4 explains the performance measurement metric and the software used for the evaluation. The experimental results are summarized in section 5. Section 6 concludes the paper.

2. Database

Three sets of data are provided in the challenge for training and development of speaker diarization (SD) systems. The first set is about 440 hours unlabeled broadcast news recordings. The second one is about 75 hours automatically labeled with another SD system. The last dataset is about 16 hours of human-revised labeled data. The final evaluation data set is also about 16 hours recordings from other channels. This data are collected from RTVE2018 [1], Aragon Radio, and 3/24 TV channel databases. The details are described as follows.

2.1. Unlabeled Data

The train partition of RTVE2018 [1] is a collection of TV shows drawn from diverse genres and broadcast by RTVE from 2015 to 2018. This partition is unlabeled and can be used for any evaluation task in Albayzin2018 [1]. The titles, duration and content of the shows included on the RTVE2018 database can be found in [1].

2.2. Automatically Labeled Data

This partition consists of the Aragon Radio, and 3/24 TV channel databases. The Aragon Radio database, which was donated by the Corporación Aragonesa de Radio y Televisión (CARTV), consists of around 20 hours of the Aragon Radio broadcast. About 35% of audio recordings contain music along with speech, 13% is noise along with speech and 22% is speech alone.

The 3/24 TV channel is a Catalan broadcast news database proposed for the Albayzin2010 Audio Segmentation Evaluation [3]. The owner of the multimedia content allows its use for technology research and development. The database consists of around 87 hours of recordings in which about 40% of the time speech can be found along with noise and 15% of the time speech along with music.
The Rich Transcription Time Marked (RTTM) files containing the segment information are generated automatically by another SD system and are provided in the challenge along with the audio recordings.

2.3. Human-Revised Labeled Data

The dev2 partition of RTVE2018 [1] contains 12 audio recordings along with their human-revised RTTM files. This partition corresponds to two different debate shows, four programs (7:26 hours) of La noche en 24H, where a group of political analysts comments what has happened throughout the day, and eight programs (7:42 hours) of Milenium where a group of experts debates about a current issue. The audio recordings consist of speech, music, noise, and a combination of them.

3. EML Speaker Diarization System

We have used the EML Online speaker diarization system in which the audio recording is processed every 0.1 sec and the decision for the speaker ID is made approximately every 2 sec. In summary, every 0.1 sec, the Voice Activity Detection (VAD) algorithm decides if the current segment is speech or nonspeech. If it is nonspeech, it will be discarded otherwise the Baum-Welch statistics are computed and accumulated over speech segments until the predefined maximum speech duration is reached. Then the accumulated statistics are converted to the supervector which is further mean normalized by the UBM mean supervector. The resulting supervector is then converted to a lower dimensional speaker vector given a transformation matrix. The process of speaker vector extraction is shown in Fig. 1. Although the same feature vectors can be used for both VAD and supervectors, we have used two different ones referred to as VAD and SPK features in Fig. 1.

The speaker vector is compared with the current speaker models using cosine similarity. If the speaker vector does not belong to any speaker, based on a speaker-dependent threshold, a new model will be created. Before a new model creation, the speech segment is divided into two halves. For each half, a speaker vector is created and compared with another using cosine similarity. If the two halves are similar enough in terms of the speaker identity, the statistics are merged and the new model is created. Otherwise, each half is assigned to one of the current speaker models. In other words, a new speaker model is created only if two halves are similar enough. In the algorithm, every speaker has its own threshold which will be updated over time. All the new created speaker models are assigned a fixed starting threshold. Then this threshold is updated based on the average scores of the assigned speaker vectors over time. More details about every part of the algorithm is given as follows.

3.1. Feature Extraction and Universal background Models

Feature vectors are extracted every 10 msec with a 25 msec window. For VAD, they are 16 dimensional MFCCs along with their deltas, and for speaker vectors are 30 dimensional static MFCCs. Features are mean normalized with a 3 sec sliding window. UBM for both VAD and speaker vectors are GMMs with 64 Gaussian mixtures each. They are trained using the unlabeled dataset described in Sec. 2.1.

3.2. Voice Activity Detection

We have used the hybrid supervised/unsupervised VAD proposed in [2]. In summary, the VAD model is based on zero-order Baum-Welch statistics obtained from the UBM. Given the speech/nonspeech labels from the automatically labeled data (Sec. 2.2), a single 64 dimensional vector of zero-order statistics is obtained for each speech and nonspeech class. In the test phase, the zero-order statistics of a given audio segment is computed and compared with speech and nonspeech vectors based on the cosine similarity.

3.3. Speaker Vectors

Speaker vectors in this algorithm are obtained by the transformation of GMM supervectors. Supervectors are first transformed by a Within-Class Covariance Normalization (WCCN) matrix and then by a Linear Discriminant Analysis (LDA) matrix to lower dimensional vectors referred to as speaker vectors in this paper. The automatically labeled data (Sec. 2.2) is used to train WCCN and LDA. The whole data is divided into two non-overlapping datasets and chosen in 0.5, 1, 1.5, and 2 sec segments. For each segment one supervector is created. One dataset is used for training the WCCN and another for the LDA. We have used 300 dimensional speaker vectors in this challenge.

3.4. Scoring

The speaker vectors are compared using cosine similarity. However, the cosine score is not stable enough to make a robust decision. Therefore, we have created a set of background speaker vectors to normalize these scores effectively before decision. The background speaker vectors are obtained on the unlabeled dataset (sec. 2.1). All the speech segments are chopped into 2 sec segments and converted to 300 dimensional speaker vectors. Afterwards, the resulting speaker vectors are clustered using a two stage unsupervised clustering technique which was used to estimate the speaker labels of the background data for training the Probabilistic Linear Discriminant Analysis (PLDA) in [4].

![Figure 1: Speaker vector extraction in the EML Online speaker diarization system.](image)
The first stage of the clustering algorithm is similar to the Mean Shift based algorithm proposed in [5] and used successfully in [6]. In the second stage, the closer clusters obtained in the first stage are combined. The second stage can be iterated for a few iterations or until no further merge is possible. In both stages, speaker vectors are joined based on the cosine similarity considering a threshold which is set to 0.350 and 0.300 for stages 1 and 2, respectively. After clustering, the centroids of the top 2048 clusters with higher number of members are considered as the final background speaker vectors.

Given the background speaker vectors, we perform a semi-S-normalization on the scores before decision. The test speaker vector and the speaker models are both compared with background speaker vectors using cosine similarity. The cosine score between the test speaker vector and the speaker model is normalized once with the mean score of the top 10 closest background speaker vectors to the test speaker vector and another time with the mean score of the top 10 closest background speaker vectors to the speaker model. The final score is the average of these two normalized scores.

### 4. Performance Measurement

As in the NIST RT Diarization evaluations, the Diarization Error Rate (DER) will be used for the performance measurement in the challenge. The DER includes the time that is assigned to the wrong speaker, missed speech time and false alarm speech time.

The speaker error time is the amount of time that has been assigned to an incorrect speaker. This error can occur in segments where the number of system speakers is greater than the number of reference speakers, but also in segments where the number of system speakers is lower than the number of reference speakers whenever the number of system speakers and the number of reference speakers are greater than zero.

The missed speech time refers to the amount of time that speech is present but not labeled by the diarization system in segments where the number of system speakers is lower than the number of reference speakers.

The false alarm time is the amount of time that a system speech has been labeled by the diarization system but is not present in segments where the number of system speakers is greater than the number of reference speakers.

As defined in the challenge [7], consecutive segments of the same speaker with a silence of less than 2 sec come together and are considered as a single segment. A forgiveness collar of 0.25 sec, before and after each reference boundary, will be considered in order to take into account both inconsistent human annotations and the uncertainty about when a speaker begins or ends. Overlap regions where more than one speaker is present are also taken into account for the evaluation.

The tool used for evaluating the diarization systems is the one developed for the RT diarization evaluations by NIST md-eval-v22.pl, available in the web site of the evaluation: http://catedrar.tve.unizar.es/reto2018. The command line for the evaluation will be as follows,

```
md-eval-v22.pl -b -c 0.25 -r reference.rttm -s system.rttm (1)
```

### 5. Experimental Results

We have used the human-revised labeled data (Sec. 2.3) for the evaluation of the diarization system. It contains 12 audio recordings from two different channels with a total duration of approximately 16 hours. The audio recordings include speech, music, silence, noise, cross talks (some times more than two speakers at a same time), and speech over music.

Two different participating conditions are proposed in this challenge, a closed-set condition in which only data provided within the Albayzin evaluation can be used for training and an open-set condition in which external data can also be used for training as long as they are publicly accessible to everyone. We have participated only in the closed-set condition.

The EML speaker diarization system is primarily designed for an Online application for which the robustness, computational cost, and the response time is important. In the primary submitted system, the decision about the identity of the speakers is made every approximately 2 sec without looking at the future in the audio recording. As the algorithm divides the speaker vectors into two halves before creating a new speaker model, the resolution for the speaker change point detection is about 1 sec. However, as the response time is not important in the challenge, we can increase it to a longer time but it would correspond to losing fast speaker turns in the audio recording.

The development set (sec. 2.3) includes audio recordings from two Spanish programs, La noche en 24H and Millenium. The experimental results showed higher average DER on La noche en 24H recordings than on Millenium recordings. It could be due to longer duration of audio signals (2 hours each compared to 1 hour each for Millenium), faster speaker turns, more cross talks, more music on the background, or something else which needs more investigation.

Table 1 summarizes the average DER on each program, the total DER obtained on the entire dev2 set of RTVE2018 dataset considering the duration of audio recordings, and the total computational time used for processing all the recordings from scratch including the feature extraction. The processing is made using a single core of an Intel(R) Xeon(R) CPU @2.10GHz.

### 6. Conclusions

We used the EML Online speaker diarization system as a participation in the recent Albayzin speaker diarization evaluation. We tried to take advantage of all the unlabeled and labeled data provided in the challenge in the close-set condition. The system showed a reasonable performance on the development data with a very low computational cost.
7. Acknowledgement
We would like to thank Wei Zhou for the efficient implementation of the Online algorithm.

8. References
In-domain Adaptation Solutions for the RTVE 2018 Diarization Challenge

Ignacio Viñals, Pablo Gimeno, Alfonso Ortega, Antonio Miguel, Eduardo Lleida
ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain
{ivinalsb, pablogj, ortega, amiguel, lleida}@unizar.es

Abstract

This paper tries to deal with domain mismatch scenarios in the diarization task. This research has been carried out in the context of the Radio Televisión Española (RTVE) 2018 Challenge at IberSpeech 2018. This evaluation seeks the improvement of the diarization task in broadcast corpora, known to contain multiple unknown speakers. These speakers are set to contribute in different scenarios, genres, media and languages. The evaluation offers two different conditions: A closed one with restrictions in the resources to train and develop diarization systems, and an open condition without restrictions to check the latest improvements in the state-of-the-art.

Our proposal is centered on the closed condition, specially dealing with two important mismatches: media and language. ViVoLab system for the challenge is based on the i-vector PLDA framework: i-vectors are extracted from the input audio according to a given segmentation, supposing that each segment represents one speaker intervention. The diarization hypotheses are obtained by clustering the estimated i-vectors with a Fully Bayesian PLDA, a generative model with latent variables as speaker labels. The number of speakers is decided by comparing multiple hypotheses according to the Evidence Lower Bound (ELBO) provided by the PLDA, penalized in terms of the hypothesized speakers to compensate different modeling capabilities.

Index Terms: adaptation, diarization, broadcast, i-vector, PLDA, Variational Bayes

1. Introduction

The production of broadcast content has progressively augmented along the last years, becoming more and more necessary the tools to process and label all these new data. One of the required tasks is diarization, the indexation of some audio according to the active speaker. Hence the goal of diarization is the differentiation among the speakers by means of generic labels, leaving the identification of each speaker for further work, if necessary. Originally developed for telephone conversations, new domains such as broadcast audio and meetings are suitable to be interested in this technique, adding new challenging drawbacks not present in the original scenario.

Multiple approaches have been proposed to the diarization problem since its origins, most of them following two main strategies: The Top-Down philosophy, which obtains the correct labels by dividing an initial hypothesis with only one speaker, and the Bottom-Up strategy, which initially divides the input audio into acoustic segments containing only one speaker each, and combining them afterwards. Further information is available in [1][2]. Both philosophies need to characterize the speakers in the different parts of the audio to make any decision. For this reason diarization applies many methods developed for speaker recognition. Successful diarization systems considering these technologies are: Agglomerative Hierarchical Clustering (AHC)[3] with $\Delta BIC$ [4], streams of eigenvoices[5] re-segmented with HMMs [6], i-vectors [7] clustered with PLDA [8], [9] in [10]. Neural Networks are also contributing, firstly providing more reliable acoustic information [11] and more recently a new representation: the embeddings such as x-vectors [12].

When moving from telephone data to other scenarios, such as broadcast or meetings, new difficulties arise. Specially relevant are the estimation of the number of speakers and the domain mismatch. The first problem is caused by the presence of an unknown number of speakers in the audio. This difficulty increases if the contributions per speaker are significantly unbalanced. Our proposed solution to deal with this problem is [13], which makes use of a penalized version of the Evidence Lower Bound (ELBO) from a Variational Bayes solution, as reliability metric. Another important problem is the domain mismatch. Broadcast data consists of several shows, belonging to multiple genres and many differences in terms of locations, audio quality or postprocessing details. This large variability in the audio makes that one system is likely to lack in precision to cover the whole range of possibilities. However specific systems are also unfeasible for practical reasons. Our approach[14] combines both strategies: A single system is unsupervisedly adapted to the different shows to diarize with the same data to evaluate, obtaining the diarization labels afterwards.

The paper is organized as follows: Section 2 describes the evaluation and the available data. ViVoLab system is presented in Section 3. Section 4 is dedicated to present the obtained results. Finally, some conclusions are included in Section 5.

2. RTVE 2018 Challenge

The RTVE 2018 Challenge is part of the 2018 edition of the Albayzin [15], [16], [17], [18], [19] evaluations. These evaluations are designed to promote the evolution of speech technologies in Iberian languages. In particular, RTVE 2018 Challenge is focused on the extraction of relevant information from Broadcast data in Spanish language. This information, such as the identity of the person on screen and his speech, is intended to help describing and labeling the multimedia data for further work. To accomplish all these goals, the evaluation provides around 500 hours of shows from the Spanish Public TV corporation Radio Televisión Española (RTVE). The considered audio tries to cover the widest possible range of Spanish variability, including varieties of Spanish from Spain and Latin America. In addition to the provided audio some metadata is provided, with different levels of reliability.

The database is divided into 4 subsets, with different functionality:
3. System description

ViVoLab submission is based on the diarization system firstly developed in [10]. This system, based on a bottom-up strategy, firstly divides the input audio into acoustic homogeneous segments, clustered afterwards by means of a Fully Bayesian PLDA. Its schematic is shown in Fig 1.

In the following lines we describe in details each part of the schematic.

3.1. Feature Extraction

From the input audio $s_n(t)$ 20 MFCC features vectors are extracted, including C0 (C0-C19), over a 25 ms hamming window every 10 ms (15 ms overlap). No derivatives are considered. The obtained features are normalized according to a Cepstral Mean and Variance Normalization (CMVN). The mean and variance are estimated taking into account the whole episode to diarize.

3.2. Voice Activity Detection

Voice Activity Detection (VAD) is performed by means of a 2-layer BLSTM [21] network of 128 neurons, trained on the 3/24 database. This network estimates the speech detection in 3-second duration sequences, providing one label each 10 ms of audio. The input parametrization is an 80-element Mel Filter bank and the log energy, computed in 25 ms windows with a 10 ms advance shift. The obtained features are normalized in mean and variance according to the whole audio.

3.3. Segment Representation

The speaker change detection is carried out making use of a ΔBIC [4] analysis, modeling the hypotheses with Full Covariance Gaussian distributions. A sliding window strategy has been considered for this purpose. Each one of the obtained acoustic segments is represented by an i-vector [7], with a 256-Gaussian 100-dimension i-vector extractor exclusively trained with 3/24 TV dataset. The obtained i-vectors have centering, whitening and length normalization [22] applied.

3.4. Clustering Method

The i-vector clustering is performed by a 100-dimension Fully Bayesian PLDA [9] [10] trained with 3/24 and CARTV datasets. Its Bayesian Network is shown in Fig 2. Due to the fact that known mismatches are present in the evaluation domain considering the episode to diarize itself. Performed in the PLDA model, the adaptation requires speaker labels, which are obtained by unsupervised clustering. In this case, the proposed decomposition is:

$$P(\Phi, Y, \theta, \pi_0, \mu, V, W, \alpha) \approx q(Y)q(\theta)q(\pi_0)q(\mu)q(V)q(W)q(\alpha) \quad (1)$$

3.5. Unsupervised Adaptation

Due to the fact that known mismatches are present in the evaluation (Language regarding 3/24 dataset and media according to CARTV dataset), some losses in performance could be expected. We include an unsupervised adaptation [14] to the system. Prior to any evaluation, the out-of-domain PLDA is adapted to the evaluation domain considering the episode to diarize itself. Reassigned, the related factors ($q(\theta), q(\pi_0), q(\mu)$) are iteratively reevaluated, reassigning the speaker labels in the process.

The whole clustering step consists of two differentiated steps. First we generate an initialization assignment for the i-vectors, becoming a seed for the corresponding hidden variable $\theta$. Then all the related factors ($q(\theta), q(\pi_0), q(\mu)$) are iteratively reevaluated, reassigning the speaker labels in the process.

3.6. Speaker number estimation

The considered PLDA and its Variational Bayes solution have the ability to recombine and eliminate speakers at will, only (and strongly) depending on the initial clustering. This initial clustering determines the maximum possible clusters to create as well as the first assignment of the i-vectors to these clusters. Our solution to this problem is the consideration of multiple initializations, with a different number of speakers.
best compare the results from each hypothesis, we make use of a penalized version of the Evidence Lower Bound (ELBO) given by the PLDA model [13]. This metric, related to the likelihood, indicates how well the speaker labels represent the given data, but penalized in terms of the hypothesized number of speakers to avoid unnecessary subclustering, i.e., estimating more clusters than the real number of speakers.

4. Results

ViVoLab submission to the RTVE 2018 Diarization Challenge consists of two systems, a primary and a contrastive. Both follow the pipeline described previously, with the only difference that our primary system performs unsupervised adaptation while the contrastive does not. All the models were trained with 3/24 and CARTV data and no extra knowledge was considered, not even those provided by RTVE data.

The results obtained by the previously described configurations are exhibited in Table 1.

Table 1: DER (%) Results for Primary and contrastive 1 systems in the development set. Results presented with Ground truth VAD and our BLSTM VAD for comparison reasons. Results include the DER term as well as its contributions: Miss speech (MISS), False Alarm speech (F.A.) and Speaker Error(SPK). Overlap is considered for evaluation purposes.

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>MISS (%)</th>
<th>F.A. (%)</th>
<th>SPK(%)</th>
<th>DER(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primary</td>
<td>1.78</td>
<td>0.00</td>
<td>9.30</td>
<td>11.08</td>
</tr>
<tr>
<td>Contrastive</td>
<td>1.78</td>
<td>0.00</td>
<td>8.05</td>
<td>9.83</td>
</tr>
<tr>
<td>BLSTM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primary</td>
<td>2.51</td>
<td>1.74</td>
<td>6.69</td>
<td>10.94</td>
</tr>
<tr>
<td>Contrastive</td>
<td>2.51</td>
<td>1.74</td>
<td>9.99</td>
<td>14.24</td>
</tr>
</tbody>
</table>

An analysis of the obtained results in terms of the estimation of the number of speakers can be done. A simple analysis is available in Table 2.

5. Conclusions

The ViVoLab submission to the RTVE 2018 Challenge includes two systems that have satisfactory results when considering the development set.

According to the development set, both systems are robust enough to deal with the subset mismatches, especially those known during the training phase: Language and Media. The trained models perform well while evaluating unknown audio.

Table 2: Absolute and relative error in the estimation of the number of speakers for both the primary and contrastive systems. Error defined as \#dia - \#oracle. Ground truth (ORACLE) and BLSTM VAD conditions are studied. Analysis carried out on the development set.

<table>
<thead>
<tr>
<th>System</th>
<th>Absolute Error</th>
<th>Relative Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primary</td>
<td>12.91</td>
<td>97.87</td>
</tr>
<tr>
<td>Contrastive</td>
<td>1.83</td>
<td>16.40</td>
</tr>
<tr>
<td>BLSTM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primary</td>
<td>18.16</td>
<td>132.83</td>
</tr>
<tr>
<td>Contrastive</td>
<td>7.5</td>
<td>56.12</td>
</tr>
</tbody>
</table>

from a new domain. Besides, when moving from ground truth VAD to a noisy one, a new sort of audio mismatch is introduced: i-vectors with non-speech. In this situation the unsupervised adaptation contributes learning from the evaluation data and adapting the model to the new conditions. This adaptation makes the system not to be degraded by the new scenario. In real conditions with noisy VAD the adaptive solution obtains a 24% relative improvement respect to the non-adaptive contrastive system.

Regarding the estimation of the number of speakers, the performed analysis indicates that our systems are likely to sub-cluster, i.e., hypothesize a larger number of clusters than the real value. If some of these extra clusters are dedicated to collect strange segments, the main clusters keep pure and clean and compensate any loss in performance. Therefore some sub-clustering could help avoiding relevant mistakes (primary system vs contrastive system using ground truth VAD). However, our results also indicate that an excessive sub-clustering causes a strong degradation of the performance (contrastive system with BLSTM VAD). Further research should be done to provide a deeper understanding.

6. References


DNN-based Embeddings for Speaker Diarization in the AuDiaS-UAM System for the Albayzin 2018 IberSPEECH-RTVE Evaluation

Alicia Lozano-Diez, Beltran Labrador, Diego de Benito, Pablo Ramirez, Doroteo T. Toledano

Audias-UAM - Audio, Data Intelligence and Speech Research Group, Universidad Autonoma de Madrid (UAM), Madrid, Spain

alicia.lozano@uam.es

Abstract

This document describes the three systems submitted by the AuDiaS-UAM team for the Albayzin 2018 IberSPEECH-RTVE speaker diarization evaluation. Two of our systems (primary and contrastive 1 submissions) are based on embeddings which are a fixed length representation of a given audio segment obtained from a deep neural network (DNN) trained for speaker classification. The third system (contrastive 2) uses the classical i-vector as representation of the audio segments. The resulting embeddings or i-vectors are then grouped using Agglomerative Hierarchical Clustering (AHC) in order to obtain the diarization labels. The new DNN-embedding approach for speaker diarization has obtained a remarkable performance over the Albayzin development dataset, similar to the performance achieved with the well-known i-vector approach.

Index Terms: speaker diarization, embeddings, i-vectors, AHC

1. Introduction

The AuDiaS-UAM submission for the Speaker Diarization (SD) evaluation consisted of three different systems, two of them based on embeddings (also known as x-vectors) [1] extracted from a Deep Neural Network (DNN) trained for speaker classification, and a third one based on the classical total variability i-vector model [2].

Our systems are submitted for the closed-set condition since they are trained using the training and development datasets made available for this evaluation, briefly described in Section 2.

For all our systems, we extract frame-level features as described in Section 3. Then, using those features, we train either a DNN or an i-vector extractor in order to obtain a fixed-length representation of an audio segment (regardless its duration), as presented in Sections 4 and 5, respectively. These models are trained using a segmentation based on the reference labels (RTTM files) provided by the organizers for training and development purposes.

We kept three development files from RTVE dataset as held out set for diairization performance evaluation. For these recordings, in order to discard fragments where just music (without speech) is present, we developed a DNN-based music detector described in Section 6. Then, diarization labels are obtained by means of Agglomerative Hierarchical Clustering (AHC) performed over non-music segments. This last step is summarized in Section 7. Finally, in Section 8 we show results of the AuDias-UAM submitted systems over our development dataset.

2. Training and Development Datasets

We used the three datasets provided by the organizers for this evaluation: Aragon Radio, 3/24 TV channel and RTVE 2018 (dev2) [3]. For training, we used all data from the first two databases and 8 files (out of 12) from RTVE 2018. This set was segmented according to the time alignments specified by the RTTM files.

In order to evaluate the speaker diarization performance of our systems, we used 3 files from the RTVE 2018 development set (approximately 4 hours) not used for training.

All the audio files were down-sampled to 16kHz.

3. Feature Extraction

All our systems are based on MFCC features extracted using Kaldi [4]. Each feature vector consists of 20 MFCCs (including C0), computed every 10 ms with a 25 ms “Povey” window (default in Kaldi, similar to Hamming window).

For the i-vector system, these 20-dimensional features are normalized using cepstral mean normalization over a 3 s sliding window, and augmented with their first and second derivatives (Δ and ΔΔ), providing a final 60-dimensional feature vector.

For the DNN-embedding systems, the raw 20-dimensional MFCC feature vectors are used to feed the network without applying channel compensation or adding temporal information. However, global zero-mean and unit-variance normalization is performed over the whole training set.

4. DNN-based Embedding Systems

Two of our submitted systems are based on DNN-based embeddings [1]. An embedding is a fixed-length representation of a given utterance or audio segment learned directly by a DNN. Typically, this DNN is trained for speaker classification.

In our case, we used an architecture based on Bidirectional Long Short-Term Memory (BLSTM) recurrent neural networks similar to the one used in [5] for language recognition, whose configuration was adjusted to the available data and the speaker diarization task. The architecture (sequence-summarizing DNN) used consists on a frame-level part composed of two BLSTM layers (with 128 cells each) and a fully-connected layer of 500 hidden units. Then, a pooling layer computes the mean and standard deviation over time to the outputs of the previous layer, followed by two fully connected layers (embeddings a and b, respectively) of 50 hidden units each and a softmax output layer with 3124 output units, working on an utterance-level basis. All the layers (except the output layer) use sigmoid non-linear activation. A graphical representation of the architecture is depicted in Figure 1.

The size of the output layer (3124) corresponds to the number of speakers considered in our training dataset. However, it should be pointed out that due to the lack of actual speaker identification labels and the segmentation based on the RTTM labels for diarization, each recording was considered to have
different speakers than the rest (which is usually not the case). Then, even though two segments might be labeled as spoken by different speakers, they could belong to the same person.

The DNN was trained using stochastic gradient descent to minimize the multi-class cross-entropy criteria for speaker classification. For training purposes, the network was fed with 3 s long sequences of 20-dimensional MFCC feature vectors.

After training, embeddings were extracted for each 3 s fragment of the development and test recordings, with a shift of 0.5 s. Each segment was forwarded through the network up to the first embedding layer (embedding a), providing a 50-dimensional embedding every 0.5 s (corresponding to 3 s sequences).

This system was implemented using Keras [6]. Embeddings obtained from this system were used for the primary system and the contrastive system 1, which differ in the clustering stage as described in Section 7.

5. I-vector System

As contrastive system 2, we used the classical total variability i-vector [2] modeling.

To develop this system, an UBM of 1024 Gaussian components was trained using the 60-dimensional MFCC+Δ+ΔΔ features described in Section 3, and a 50-dimensional total variability subspace was derived from the Baum-Welch statistics of the training segments (obtained according to RTTM timestamps). The configuration was taken from previous speaker diarization systems developed in our research group. After training, each development and test recording was processed in order to obtain a stream of i-vectors every 0.5 s (as with embeddings) with a sliding window of length 3 s.

This system was implemented using Kaldi [4]. The speaker diarization was performed using clustering on top of the resulting streams of i-vectors (see Section 7).

6. Music Detection

In order to discard segments where just music was present, we developed a music/speech classifier based on DNNs [7]. This system is trained using 150 h of audio from Google Audio Set [8], a dataset consisting of 10 seconds audio segments extracted from YouTube videos. Our architecture is composed of six bidimensional convolutional neural network (CNN) layers which operate on the Mel-spectrogram of the audio signals, followed by one LSTM layer and a final fully-connected layer prior to the output layer.

The output layer has 4 output units which correspond to the classification into music, speech, speech + music and none of them. For this evaluation we used just the probabilities of belonging to the music class in order to filter out segments that contain only music. This way, just the embeddings or i-vectors corresponding to speech (or speech with music) segments were used to perform the clustering stage.

In order to extract the probability of a segment to contain music, Mel-spectrograms corresponding to test recordings have been computed and split into a stream of 10 second segments to fit the input size of the music/speech classifier. The separation between consecutive segments is 0.5 s in order to identify each Mel-spectrogram with an embedding or i-vector in the stream.

7. Agglomerative Hierarchical Clustering

In order to obtain the speaker diarization labels, we used Agglomerative Hierarchical Clustering (AHC) over the resulting stream of either embeddings or i-vectors (depending on the system) for a given development or test recording. Embeddings or i-vectors corresponding to music segments according to our music detector were discarded previously to the clustering step. This stage was implemented in Python using the scikit-learn toolbox [9].

Thus, AHC is applied to the resulting sequence of vectors corresponding to a specific audio file. We used cosine distance for i-vectors and euclidean distance for embeddings. The number of clusters is controlled by the threshold of the distance to merge clusters, whose value was optimized on the development set. The linkage method used was the average of the vectors. This clustering was applied once for the contrastive systems.

However, for the primary system we applied first AHC over the whole set of vectors with a lower threshold to allow a bigger number of clusters, and then, a second AHC stage was applied to group the centroids of the previous clusters. This was done in order to help the clustering grouping speaker identities instead of vectors similar due to their closeness in time. The centroids were computed as the mean vector over all the points labeled as belonging to the same cluster in the first AHC stage.

Finally, for all the systems, we post-processed the clustering labels by filtering out clusters that grouped less than 10 s of audio. This was done in order to reduce false alarm in terms of clusters that do not group a different speaker but segments further than the chosen threshold.

8. Development Results

Table 1 shows the results obtained by our systems in our development set (three files from RTVE 2018 dev 2 partition).

In our development set, both systems based on embeddings obtained similar results, especially in terms of missed and false alarm speaker time. The difference in these two metrics with respect to the third system (based on i-vector) is also not significant, while the performance differs mainly due to the speaker error time. This might be related to the system settings and thresholds selected for the clustering, which would merge different speakers into one cluster or vice-versa.

Even though the i-vector system obtained better performance in our development dataset than the embedding-based systems, we submitted as primary one of these systems due to the novelty of this technique with respect to the well-known i-vectors for speaker diarization and related tasks.
Table 1: Performance of the Audias-UAM submission for the Albayzin IberSPEECH-RTVE speaker diarization evaluation over the development dataset (approximately 4 hours, 3 different recordings from 2 different shows. The performance is shown in % of scored speaker time.

<table>
<thead>
<tr>
<th>System</th>
<th>Missed</th>
<th>False Alarm</th>
<th>Speaker Error</th>
<th>DER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embeddings + double AHC (primary)</td>
<td>2.2</td>
<td>1.2</td>
<td>14.6</td>
<td>18.02</td>
</tr>
<tr>
<td>Embeddings + simple AHC (contrastive 1)</td>
<td>2.1</td>
<td>1.2</td>
<td>15.3</td>
<td>18.65</td>
</tr>
<tr>
<td>i-vectors + simple AHC (contrastive 2)</td>
<td>2.9</td>
<td>1.3</td>
<td>12.9</td>
<td>17.22</td>
</tr>
</tbody>
</table>

9. Acknowledgements

This work was supported by project DSSL: Redes Profundas y Modelos de Subespacios para Detección y Seguimiento de Locutor Idioma y Enfermedades Degenerativas a partir de la Voz (TEC2015-68172-C2-1-P), funded by Ministerio de Economía y Competitividad, Spain and FEDER.

10. References


CENATAV Voice-Group Systems for Albayzin 2018 Speaker Diarization Evaluation Campaign

Edward L. Campbell Hernández, Gabriel Hernández Sierra, José R. Calvo de Lara
Voice Group, Advanced Technologies Application Center, CENATAV, Havana, Cuba
eecampbell@cenatav.co.cu, gsierra@cenatav.co.cu, jcalvo@cenatav.co.cu

Abstract

Usually, the environment to record a voice signal is not ideal and, in order to improve the representation of the speaker characteristic space, it is necessary to use a robust algorithm, thus making the representation more stable in the presence of noise. A Diarization system that focuses on the use of robust feature extraction techniques is proposed in this paper. The presented features (such as Mean Hilbert Envelope Coefficients, Medium Duration Modulation Coefficients and Power Normalization Cepstral Coefficients) were not used in other Albayzin Challenges. These robust techniques have a common characteristic, which is the use of a Gammatone filter-bank for dividing the voice signal in sub-bands as an alternative option to the classical Triangular filter-bank used in Mel Frequency Cepstral Coefficients. The experiment results show a more stable Diarization Error Rate in robust features than in classic features.

Index Terms: Speaker Diarization, Robust feature extraction, Mean Hilbert Envelope Coefficients, Albayzin 2018 SDC

1. Introduction

This is the first participation of the CENATAV Voice Group in the Albayzin Challenges, participating in the Speaker Diarization Challenge (SDC) task and developing a Diarization System focuses in robust feature extraction. A Speaker Diarization System allows identifying "Who spoke when?" on an audio stream, which has been of interest for the scientific community since the last century, with the emergence of the first works on speaker segmentation and clustering [1][2]. The diarization can be used as a stage that enriches and improves the results of other systems, for example: a Rich Transcription System uses the diarization for adding the information about who is speaking to the speech transcription, or a Speaker Recognition System uses it when the test signal has several speakers, so diarization allows finding the segments from the test signal with only one speaker [3].

An issue of the diarization is the environment where the speech is recorded, because noise is a natural condition in real applications. The proposed system is focused on robust feature extraction techniques for improving the results in a real application. Robust techniques as Mean Hilbert Envelope Coefficients (MHEC), Medium Duration Modulation Coefficients (MDMC) and Power Normalization Cepstral Coefficients (PNCC) are analysed.

The system was mainly developed on S4D tool [4], with the following structure: robust feature extraction, segmentation (gaussian divergence and Bayesian Information Coefficient), speech activity detection (Support Vector Machine), clustering (Hierarchical Agglomerative Clustering) and the last stage is the Re-segmentation (Viterbi algorithm). A system description is done in the next sections.

2. Robust Feature Extraction

A feature is robust when it has a stable effectiveness both in controlled or uncontrolled environment (noise, reverberation, etc.), being the second condition the most common in the practice [5], so the use of a feature with this characteristic is relevant in real applications. A brief description of several robust feature extraction techniques is provided in this section. These techniques use a gammatone filter-bank (see Fig. 1), the design of which was based on Patterson’s ear model [6], defining the impulse response at the channel $i$ for the equation 1.

$$h(t)_i = \frac{\gamma \ast t^{-1} \ast \cos(2\pi \ast f_{c_i} \ast t + \theta)}{\exp(-2\pi \ast erb_i \ast t)}, \quad (1)$$

where:

- $\gamma$: amplitude.
- $\tau$: filter order.
- $erb$: equivalent rectangular bandwidth.
- $f_{c_i}$: center frequency at the channel $i$.
- $\theta$: phase.

The next gammatone filter parameters were set in the proposed system, following Glasberg and Moore’s recommendation [6], where:

- $f_{c_i} = -({EarQ \ast minB}) + \frac{(f_{max} \ast EarQ + minB)}{\exp(\text{EarQ} \ast \text{minB})}$
- $erb = \left(\frac{f_{c_i}}{EarQ} + minB\right)^{1/\gamma}$
- $EarQ = 9.26449$
- $minB = 24.7$

Figure 1: Gammatone filter-bank of 40 dimension
Earq is the asymptotic filter quality at high frequencies and \( \min B \) is the minimum bandwidth for low frequencies channels. The parameters \( \theta, \gamma, \tau \) were set to 0, 1 and 4 respectively.

2.1. Mean Hilbert Envelope Coefficients

A gammatone filter modulates a signal in amplitude and frequency [7], and to demodulate the output signal is a way for recovering the information transmitted. Mean Hilbert Envelope Coefficients (MHEC) extract this information by applying Hilbert Transform for estimating the analytic signal, separating the AM component from the modulated signal and assuming that the FM component does not exist [7]. The extraction process is shown in the figure 2.

![Figure 2: Extraction process of MHEC](image)

A low-pass filtering is done on the estimated envelope in order to eliminate rapid changes of the signal, which are usually attributed to noise [8].

2.2. Medium Duration Modulation Coefficients

Medium Duration Modulation Coefficients (MDMC) is called "medium duration" because it employs a window of 52 ms, a bigger length than the traditional windowing of 20-30 ms. However, a length of 25 ms is used in this proposal for efficiency purposes. This feature takes the same approach that MHEC. It demodulates the gammatone output signal [9]. However, in MDMC is assumed that the FM component exists, applying the Teager’s Operator (TEO) for estimating the AM component [9]. TEO is a non-linear operator that tracks the energy of a signal, which is a function of the amplitude and frequency [10]. The figure 3 shows the process for extracting the MDMC.

![Figure 3: Extraction process of MDMC](image)

2.3. Power Normalization Cepstral Coefficients

Power Normalization Cepstral Coefficients (PNCC) does not apply any technique for demodulating or separating the AM-FM component at the gammatone filter-bank output, rather the power at each sub-band is computed and transformed into the cepstral domain. There are two main approaches of PNCC, the short and medium time approaches, being the first the approach used in this paper. For more details about this technique, see [11]. The figure 4 shows the process for computing PNCC.

![Figure 4: Extraction process of PNCC](image)

3. Proposed Diarization System

The proposed Diarization System (see figure 5) has six stages:

1. Robust feature extraction: based on MHEC, MDMC and PNCC as test features.
2. Gaussian Divergence: a sliding window of 2.5 seconds is applied along the signal, two Gaussians with diagonal covariance are computed from left and right half of the window in each shift. A change point of acoustic classes (different speakers, music and noise) is detected on a middle point of the window when the gaussian divergence score reaches a local maximum.
3. Bayesian Information Coefficient (BIC): Segmentation stage for decreasing the over-segmentation originated in the previous stage. A BIC approach is used for comparing consecutive segments and merging the couple segments that belong to the same acoustic class.
4. Support Vector Machine (SVM): A SVM [12] is applied as speech/non-speech classifier of the resulting segments from the previous stage. The non-speech segments are deleted.
5. Hierarchy Agglomerative Clustering (HAC): The segments that belong to the same speaker are clustered for a traditional HAC, using BIC as comparative measure.
6. Re-segmentation: a HMM is trained on the whole signal and a Viterbi re-segmentation is done for redefining the change points.

4. Experiment
The tested robust feature extraction, LFCC and LPCC algorithms are self-implementations, while MFCC implementation is from SIDEKIT tool [13] and the SVM from pyAudioAnalysis tool [12]. The Gaussian Divergence, BIC, HAC and Re-segmentation algorithms are from S4D tool [4]. The proposed systems were submitted at closed condition and they do not use training data, with the exception of SVM, for which a portion of Albayzin SDC 2016 Database was used. The experiment was developed on RTVE-2018 SDC Development Database, tuning the default thresholds of BIC and HAC in S4D, and comparing robust (MHEC, MDMC, PNCC) and classic (MFCC, LFCC, LPCC) feature extraction techniques. The tables 1 and 2 show the best configurations at each feature. The pre-emphasis (0.97), length window (0.025 sec), shift window (0.01 sec), compression (logarithmic) and normalization (Cepstral Mean Normalization) are equal in each feature.

Table 1: Robust features configuration

<table>
<thead>
<tr>
<th>Parameters</th>
<th>MHEC</th>
<th>MDMC</th>
<th>PNCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter-bank dimension</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Bandwidth (Hz)</td>
<td>0 - 7000</td>
<td>0 - 7000</td>
<td>0 - 7000</td>
</tr>
<tr>
<td>Cepstral coefficients</td>
<td>12 + Δ + ΔΔ</td>
<td>12 + Δ + ΔΔ</td>
<td>15 + Δ + ΔΔ</td>
</tr>
</tbody>
</table>

Table 2: Classic features configuration

<table>
<thead>
<tr>
<th>Parameters</th>
<th>MFCC</th>
<th>LFCC</th>
<th>LPCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter-bank dimension</td>
<td>40</td>
<td>40</td>
<td>-</td>
</tr>
<tr>
<td>Bandwidth (Hz)</td>
<td>0 - 7500</td>
<td>0 - 7500</td>
<td>-</td>
</tr>
<tr>
<td>Prediction order</td>
<td>-</td>
<td>-</td>
<td>60</td>
</tr>
<tr>
<td>Cepstral coefficients</td>
<td>19 + Δ</td>
<td>19 + Δ</td>
<td>19 + Δ + ΔΔ</td>
</tr>
</tbody>
</table>

5. Results
The figure 6 presents the results of the proposed diarization system, using several feature extraction techniques and separating these result for each TV-show of the SDC Development Database, “La noche en 24H” and “Millennium”. The objective is to find a stable feature, where the Diarization Error Rate (DER) changes little between different signals. The figure 6 shows that the robust features are more stable than the classics features and MHEC is the most stable between these robust features. The general system proposed is showed in figure 5, and the primary and contrastive systems are based in the general system, using the following features:

- **Primary system**: MHEC. The most stable feature and the best performance on the “La noche en 24H” group.
- **Contrastive system-1**: MDMC. The best general performance.
- **Contrastive system-2**: LFCC. The best performance on the “Millennium” group.

Despite MFCC is the most used feature extraction in speech processing, this algorithm does not present a better performance than the robust features proposed in this paper.

Figure 6: Diarization Error Rate of the proposed system by feature.

The computational cost was computed in terms of real-time factor. This measure represents the necessary time for processing a second of signal. The experiment was done on Intel Core i3-3110M CPU 2.40 GHz X 4 with 3.7 GB of memory. The computational cost of each system submitted is shown in the table 3, being the system with LFCC the most efficient.

Table 3: Real-time factor of each system submitted

<table>
<thead>
<tr>
<th></th>
<th>Primary</th>
<th>Contrastive-1</th>
<th>Contrastive-2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.30</td>
<td>0.23</td>
<td>0.04</td>
</tr>
</tbody>
</table>

6. Conclusion
This paper was reported by the CENATAV Voice-Group, and submitted for Albayzin 2018 Speaker Diarization Challenge. The main proposal was based on robust feature extraction, using MHEC as primary algorithm of feature extraction, with the objective of providing a stable system. The Diarization Error Rate of the primary system on development data was 15.23 %, with a real-time factor of 0.3, being suitable for a real application.

7. Acknowledgments
Finally, we would like to thank the University of Zaragoza, in the person of the professor Eduardo Lleida Solano, for his support during the experiment development. And to thank Wilma Recio for her help during the redaction of this paper.

8. References


The Intelligent Voice System for the IberSPEECH-RTVE 2018
Speaker Diarization Challenge

Abbas Khosravani, Cornelius Glackin, Nazim Dugan, Gérard Chollet, Nigel Cannings
Intelligent Voice Limited, St Clare House, 30-33 Minories, EC3N 1BP, London, UK

Abstract

This paper describes the Intelligent Voice (IV) speaker diarization system for IberSPEECH-RTVE 2018 speaker diarization challenge. We developed a new speaker diarizer built on the success of deep neural network based speaker embeddings in speaker verification systems. In contrary to acoustic features such as MFCCs, deep neural network embeddings are much better at discerning speaker identities especially for speech acquired without constraint on recording equipment and environment. We perform spectral clustering on our proposed CNN-LSTM-based speaker embeddings to find homogeneous segments and generate speaker log likelihood for each frame. A HMM is then used to refine the speaker posterior probabilities through limiting the probability of switching between speakers when changing frames. We present results obtained on the development set (dev2) as well as the evaluation set provided by the challenge.

Index Terms: speaker diarization, CNN, LSTM, IberSPEECH-RTV, speaker embedding

1. Introduction

Speaker diarization is the task of marking speaker change points in a speech audio and categorizing segments of speech bounded by these change points according to the speaker identity. Speaker diarization which is also referred to as who speaks when, is an important front-end processing for a wide variety of applications including information retrieval and automatic speech recognition (ASR).

The speaker diarization performance largely varies by the application scenario including the broadcast news audio, interview speech, meetings audio (with distance microphone), pathological speech, child speech, and conversational telephone speeches. Each domain presents a different quality of the recordings (bandwidth, microphones, noise), types of background sources, number of speakers, duration of speaker turns, as well as the style of the speech. Apart from its similarity to speaker identification in classifying homogeneous segments as spoken by the same or different speakers, each domain presents unique challenges to speaker diarization. Since there is no prior knowledge regarding the speakers involved in an audio speech, the number of speakers as well as the approximate speaker change points need also to be addressed. Moreover, speech segments may be of very short duration, making i-vectors as the most common speaker representation not an appropriate option [1].

The most widely used approach in speaker diarization involves: (1) speech segmentation, where speech activity detection (SAD) or speaker change point detection is used to find rough boundaries around each speaker’s regions of speech; (2) segmentation, where same speaker segments as well as the number of speakers will be determined; and finally (3) re-segmentation, where the boundaries are further refined to produce the diarization results. The first stage intends to divide the speech into short segments of a few seconds with a single dominant speaker. Thus, the quality of the speech activity detection plays an important role in the performance [2, 3]. Using word boundaries generated by an Automatic Speech Recognition (ASR) system to produce homogeneous short segments has been proposed in [4]. Neural-network based approaches has also been investigated in [5]. The basic principle of detecting speaker turn is to use a short duration window of a few seconds and use a similarity measure to decide whether a speaker change has occurred or not. This window will then slides frame by frame over the entire audio to mark all the potential speaker change points. This window needs to be long enough to include speaker identifying information (usually 1-2 seconds long). The most common measures for change detection includes, Bayesian Information Criterion (BIC) [6], local Gaussian Divergence [7] and more recently deep neural networks [8, 9, 10].

In the clustering stage, features extracted from same speaker segments should ideally go to one cluster. These features could be the basic MFCCs, speaker factors [11], i-vectors [12, 13], or more recently d-vectors [14]. For many years, i-vector based systems have been the dominating approach in speaker verification [15]. However, thanks to the recent progress of deep neural networks, neural network based audio embeddings (d-vector) could significantly outperforms previously state-of-the-art techniques based on i-vectors, especially, on short segments [16, 17, 18, 19, 20]. The LSTM recurrent neural network [21] has been incorporated to produce speaker embeddings for the task of speaker verification [19] and later for speaker diarization [14]. In this work, we explore a similar text-independent d-vector based approach to speaker diarization. The proposed speaker embeddings using deep neural networks at the frame level is used for the task of speaker diarization.

The predominant approach for clustering is Agglomerative Hierarchical Clustering (AHC) [6] where a stopping criteria could be used to determine the number of clusters. Other popular clustering approaches include Gaussian Mixture Models (GMMs) [22], mean-shift clustering [23] and spectral clustering [13, 22, 14]. We found the effectiveness of spectral clustering algorithm which relies on analyzing the eigen-structure of an affinity matrix in our proposed diarization framework.

In the final stage, the initial rough boundaries are then refined. This is usually done using Viterbi algorithm at the acoustic feature level. For each cluster a Gaussian Mixture Model (GMM) is estimated to calculate posterior probabilities of speakers at the frame level and the process usually iterates until convergence. The Viterbi re-segmentation on raw MFCC features was found to be effective [11]. Re-segmentation in a factor analysis subspace has also been investigated in [24, 11].
where Variational Bayes (VB) system [25] proved to be the most effective. This approach implicitly performs a soft speaker clustering in a way which avoids making premature hard decisions. An extension to this was proposed at the 2013 Center for Language and Speech Processing (CLSP) Summer Workshop at Johns Hopkins University, where temporal continuity was modeled by an HMM. This extension will constrain speaker transitions and defines the speaker posterior probabilities [24]. In this paper, we intend to perform diarization with the help of deep neural network speaker embeddings. The log likelihoods for frame-level speaker embedding is estimated using a spectral clustering algorithm. In another way, HMM will serve as the re-segmentation for the spectral clustering algorithm.

The remainder of this paper is organized as follows: We outline the specific architecture in our proposed diarization system in Section 2 which includes CNN-LSTM based speaker embedding extraction, clustering and re-segmentation. Experimental results on the development set of the IberSPEECH-RTVE 2018 speaker diarization challenge are presented in Section 3. We will then conclude the paper in Section 5.

2. Diarization Framework

In this section, we review the key sub-tasks used to build current speaker diarization system based on DNN embeddings. The flowchart of our diarization system is shown in Fig. 1.

2.1. Acoustic Features

For speech parameterization we used 20-dimensional Mel-Frequency Cepstral Coefficients (MFCCs). These features are extracted at 8kHz sample frequency using Kaldi toolkit with 25 ms frame length and 10 ms overlap. For each utterance, the features are centered using a short-term (3s window) cepstral mean and variance normalization (ST-CMVN).

2.2. Speaker Embeddings

The i-vector based systems have been the dominating approach for both speaker verification and diarization applications. However, with the recent success of deep neural networks, a lot of efforts have been made into learning fixed-dimensional speaker embeddings (d-vectors) using an end-to-end network architecture that could be more effective relative to i-vectors on short segments [26, 20, 19, 27]. We employed a generalized end-to-end model using a convolutional neural networks (CNNs) and LSTM recurrent neural network. The network architecture is shown in Fig 2. It consists of two CNN layers and two dense layers at frame level followed by a bi-directional LSTM and two dense layers at utterance level. The LSTM layer maps a sequence of input feature vectors into an embedding vector. The output of the LSTM layer is then followed by two more dense layer and a length-normalization layer to produce a fixed dimensional representation for the input segment. Training is based on processing a large number of utterances in the form of a batch that contains N speakers and M utterances each. Each utterance could be of arbitrary duration. But to train the network in batch, they need to be of the same duration. We used variable length speech segments ranging from 10-20 seconds and construct batches with 30 speakers, each having 10 different segments. In the loss layer, a generalized end-to-end (GE2E) loss [19] builds a similarity matrix that is defined based on the cosine similarity between each pair of input utterances. During the training, we want the embedding of each utterance to be similar to the centroid of all that speaker’s embeddings, while at the same time, far from other speakers’ embeddings. A detailed description of GE2E training can be found at [19].

During evaluation, for every test utterance we apply a sliding window of 5 seconds (500 frames) with 80 percent overlap (we would have an speaker embedding every 100 frames). We compute the d-vector for each window. No speech activity detection has been used for this processing. Finally, a principle component analysis (PCA) is incorporated to reduce the dimensions of the resulting length-normalized embeddings (we used 8 dimension in our experiments) so as to be ready for clustering.

2.3. Clustering

We employed a spectral clustering which is able to handle unknown cluster shapes. It is based on analyzing the eigenstructure of an affinity matrix. A more detailed analysis of the algorithm is presented in [28]. We used an Euclidean distance measure to form a nearest neighbor affinity matrix on the frame-level embeddings. To estimate the number of clusters a simple heuristic based on the eigenvalues of the affinity matrix is used [13]. To mitigate the computational complexity of the spectral clustering, especially when the number of frames are too large, we can employ sampling at a specific rate.

To estimate the number of speakers, a simple heuristic based on the Calinski & Harabasz criterion [29] has been incorporated. We cluster d-vectors using K-means clustering algorithm using different value of cluster number and choose the one that maximize Calinski & Harabasz score.

2.4. Re-segmentation

The clustering algorithms are typically followed by a re-segmentation algorithm that refines the speaker transition
boundaries. This could be either in the feature space like MFCC or in the factor analysis subspace [24]. Speaker diarization in factor analysis space allows us to take advantages of speaker specific information. By contrast, lower-level acoustic features such as MFCCs are not quite as good for discerning speaker identities, but can only provide sufficient temporal resolution to witness local speaker changes. The proposed framework for diarization provides a stronger speaker representation at the frame level. As a result, when combined with a HMM to refine the speaker posterior probabilities through limiting the speaker transitions [24], the system is able to detect speaker change points. The speaker log likelihoods for the HMM are computed by the spectral clustering algorithm as described in section 2.3.

Table 1: DER (%) on the development data (dev2) as well as evaluation data of the IberSPEECH-RTVE 2018 speaker diarization challenge.

<table>
<thead>
<tr>
<th>Data</th>
<th>DER(%)</th>
<th>Err(%)</th>
<th>FA(%)</th>
<th>Miss(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev2</td>
<td>15.96</td>
<td>10.5</td>
<td>3.6</td>
<td>1.8</td>
</tr>
<tr>
<td>eval</td>
<td>30.96</td>
<td>25.2</td>
<td>4.8</td>
<td>0.9</td>
</tr>
</tbody>
</table>

3. Experiments

3.1. Training Data

Switchboard corpora (LDC2001S13, LDC2002S06, LDC2004S07, LDC98S75, LDC99S79) and NIST SRE 2004-2010 which consists of conversational telephone and microphone speech data at 8kHz sample frequency from around 5k speakers were used for training the system. Augmentation increases the amount and diversity of the existing training data. Our strategy employs additive noises and reverberation. Reverberation involves convolving room impulse responses (RIR) with audio. We use the simulated RIRs described in [30], and the reverberation itself is performed with the multi-condition training tools in the Kaldi ASpIRE recipe [31]. For additive noise, we use the MUSAN dataset, which consists of over 900 noises, 42 hours of music from various genres and 60 hours of speech from twelve languages [32]. Both MUSAN and the RIR datasets are freely available from http://www.openslr.org. We use a 3-fold augmentation that combines the original “clean” training list with two augmented copies [33]. To augment a recording, we choose between one of the following randomly:

- **babble**: Three to seven speakers are randomly picked from MUSAN speech, summed together, then added to the original signal (13-20dB SNR).
- **music**: A single music file is randomly selected from MUSAN, trimmed or repeated as necessary to match duration, and added to the original signal (5-15dB SNR).
- **noise**: MUSAN noises are added at one second intervals throughout the recording (0-15dB SNR).
- **reverb**: The training recording is artificially reverberated via convolution with simulated RIRs.

3.2. Performance Metrics

We measured performance with Diarization Error Rate (DER), the standard metric for diarization. It is measured as the total percentage of reference speaker time that is not correctly attributed to a speaker. More concretely, DER is defined as:

\[
\text{DER} = \frac{FA + MISS + ERROR}{\text{TOTAL}}
\]

where \(FA\) is the total system speaker time not attributed to a reference speaker, \(MISS\) is the total reference speaker time not attributed to a system speaker, and \(ERROR\) is the total reference speaker time attributed to the wrong speaker. Like the traditional conventions used in evaluating diarization performance [11], a forgiveness collar of 0.25 seconds will be applied before and after each reference boundary prior to scoring. The DER is reported based on the NIST RT Diarization evaluations [34].

4. Results

The IberSPEECH-RTVE 2018 Speaker Diarization is a new challenge in the ALBAYZIN evaluation series. This evaluation consists of segmenting broadcast audio documents according to different speakers and linking those segments which originate from the same speaker. We used two Intel Xeon CPU (E5-2670 @ 2.60GHz and 8 cores), 64G of DDR3 memory, 400G disk storage and an NVIDIA TITAN X GPU (12G of memory) to train the network. Keras API with tensorflow backend has been used for system development. Training takes almost a week to process around half a million segments of 10-20 seconds long. To process a single 20 minute recording the system execution times is around 7 seconds. We report the performance of our proposed diarization framework on the development set (dev2) using the provided speaker marks and also the result of the submitted system on the evaluation set in Table 1. Our system was trained on publicly accessible data which totally differ from both the development and evaluation data (open-set condition). The results indicate the effectiveness of the proposed approach on challenging domains.
5. Conclusion
The IberSPEECH-RTVE 2018 Speaker Diarization has proven to be a highly challenging contest especially in the detection of the number of speakers and dealing with background noise. We have presented our system and reported the results on the development set as well as the evaluation set of the challenge. We found deep neural network embeddings much better at discerning speaker identities especially for speech acquired without constraint on recording equipment and environment. Our strategy to employ additive noises and reverberation for data augmentation plays an important role in the success of our system on challenging domain. We will perform research on the evaluation set once the labels are released to gain insights on the real effects of the approaches presented in the paper.

6. Acknowledgement
The research leading to the results presented in this paper has been (partially) granted by the EU H2020 research and innovation program under grant number 769872.

7. References


JHU Diarization System Description

Zili Huang, Paola García, Jesús Villalba, Daniel Povey, Najim Dehak
Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA
{hzili1,pgarci27,jvillal7}@jhu.edu, dpovey@gmail.com, ndehak3@jhu.edu

Abstract

We present the JHU system for Iberspeech-RTVE Speaker Diarization Evaluation. This assessment combines Spanish language and broadcast audio in the same recordings, conditions in which our system has not been tested before. To tackle this problem, the pipeline of our general system, developed entirely in Kaldi, includes an acoustic feature extraction, a SAD, an embedding extractor, a PLDA and a clustering stage. This pipeline was used for both, the open and the closed conditions (described in the evaluation plan). All the proposed solutions use wide-band data (16KHz) and MFCCs as their input. For the closed condition, the system trains a DNN SAD using the Albayzin2016 data. Due to the small amount of data available, the i-vector embedding extraction was the only approach explored for this task. The PLDA training utilizes Albayzin data followed by an Agglomerative Hierarchical Clustering (AHC) to obtain the speaker segmentation. The open condition employs the DNN SAD obtained in the closed condition. Four types of embeddings were extracted, x-vector-basic, x-vector-factored, i-vector-basic and BNF-i-vector. The x-vector-basic is a TDNN trained on augmented Voxceleb1 and Voxceleb2. The x-vector-factored is a factored-TDNN (TDNN-F) trained on SRE12-micphn, MX6-micphn, VoxCeleb and SITW-dev-core. The i-vector-basic was trained on Voxceleb1 and Voxceleb2 data (no augmentation). The BNF-i-vector is a BNF-posterior i-vector trained with the same data as x-vector-factored. The PLDA training for the new scenario uses the Albayzin2016 data. The four systems were fused at the score level. Once again, the AHC computed the final speaker segmentation.

We tested our systems in the Albayzin2018 dev2 data and observed that the SAD is of importance to improve the results. Moreover, we noticed that x-vectors were better than i-vectors, as already observed in previous experiments.

Index Terms: speaker diarization, DNN, SAD, i-vectors, x-vectors, PLDA

1. Introduction

We present JHU’s speaker diarization system for Iberspeech Diarization Evaluation. Our main goal is to test our current diarization system in other databases and possible scenarios. We provide a system solution for the open and closed condition scenario, using Kaldi. We follow a basic pipeline with specific characteristics for each scenario. The pipeline for our system is described as follows.

- Audio feature extraction
- Speech activity detection (SAD)
- Embedding extraction
- PLDA
- Score fusion (if possible)
- Clustering

Each of these items will be discussed in detail in the following sections.

2. Datasets

For the closed condition we only used the Albayzin2016 to train the SAD, the Universal Background Model (UBM), the i-vector extractor and PLDA models.

The datasets used for training the open condition are:

- Data set 1: Voxceleb1 and Voxceleb2 data (no augmentation). [1][2]
- Data set 2: SRE12-micphn, MX6-micphn, VoxCeleb and SITW-dev-core.
- Data set 3: Voxceleb1 and Voxceleb2 with augmentation.
- Data set 4: Fisher database (only database not in 16KHz)
- Albayzin2016: In-domain data set used in previous evaluations, which includes Aragon Radio database (20 hours) and 3/24 TV channel database (87 hours).
- Albayzin2018: RTVE broadcast data, which includes train, dev1, dev2 and test. The speaker diarization label is only offered for dev2, so the dev2 part is used for development purposes and tuning our system.

3. Feature Extraction and Speech Activity Detection

All the systems employ wide-band data (16KHz). MFCCs were extracted for a 25ms window and 10ms frame rate. For the open condition, the MFCC feature dimension of x-vector-basic, x-vector-factored, i-vector-basic and BNF-i-vector is 30, 40, 24, 40 respectively. For the closed condition, the MFCC configuration is the same as i-vector-basic.

3.1. Speaker Activity Detection

After computing the MFCCs for each case, the system trained a TDNN SAD model on the Albayzin2016 labeled data following the Aspire recipe in Kaldi [3]. The network consists of 5 TDNN layers and 2 layers of statistics pooling[4]. The overall context of the neural network is around 1s, with around 0.8s of left context and 0.2s of right context. This approach is suitable for our purposes since it can include a wider context not affecting the number of parameters. For this special case, we trained the DNN with two classes: speech and non-speech. The speech segments include both the clean voice and the voice with noises. Other parts of the audio are considered as non-speech, which may include music, noise and silence. A simple Viterbi decoding on a HMM with duration constraints of 0.3s for speech and 0.1s for silence is used to get speech activity labels for the test set.

1 except set 4 that is in 8 KHz to train bottleneck DNN
data recordings. The energy based SAD was also tried for our experiment, but the results were worse overall.

4. Embeddings

We computed two different sets of embeddings depending on the condition. For the closed condition we focused on the i-vectors. This approach computes i-vectors in the traditional way; it trains a T-matrix with Albayzin2016-only data. Afterwards, we obtained the i-vectors for the Albayzin2018 dev2 and test set. We tried other DNN possibilities, but due to the few amount of data available the results were not promising.

For the open condition we examined four types of embeddings. The i-vectors-basic, trained on data set 3, obtained base-line results for Albayzin2018 dev2 and test. These i-vectors are of dimension 400.

The BNF-i-vectors (of dimension 600) use the bottleneck feature computed from data set 2 to refine the GMM alignments. The rest of the i-vector pipeline remains the same; the T-matrix was also trained on data set 2.

We explored two types of DNN based embedding architectures. The first one, the default Kaldi recipe for Voxceleb, is a TDNN for x-vector-basic [5, 6]. In this approach, each MFCC frame is passed through a sequence of TDNN layers. Then, a pooling layer accounts for the utterance level process and computes the mean and standard deviation of the TDNN output over time in a pooling layer. This intermediate representation, known as embedding, is projected to a lower dimension (512 in this case). The DNN output are the posterior probabilities of the training speakers. The objective function is cross entropy. We employed data set 3 for training the TDNN. The augmentation is performed as described in [7] using MUSAN noises.

For the second x-vector approach the pre-pooling layers are changed to factorized TDNNs (TDNN-F) with skip connections [8]. This new architecture reduces the number of parameters in the network by factorizing the weight matrix of each TDNN layer into the product of two low-rank matrices. The first factor is forced to be semi-orthogonal that will prevent the lost of information when projecting from high to low dimension. As in other architectures, skip connections are an option for this TDNN-F. Some input layers receive as input the output of the previous layer and other prior layers. The best solution so far is to have skip connection between low-rank interior layers in the TDNN-F. The x-vectors are of dimension 600.

5. PLDA and Score Fusion

For the closed condition we observed that the number of speakers estimated by the current approach was very high for the Albayzin2018 dev2 set. We decided to use PCA as in [9]. With this tuning strategy the system was able to take into account every recording for PCA rotation, instead of only the global PCA. This strategy also maintained the number of speaker in a desirable range.

For the open condition, we used the traditional PLDA workflow, and the PLDA was trained on the Albayzin2016 data. We obtained 4 different types of scores that addressed the four type of embeddings. We fused the four systems with equal weights.

6. Clustering

The system performed an Agglomerative Hierarchical Clustering (AHC) to obtain a segmentation of the recordings following Albayzin diarization recipe [3]. To obtain an accurate estimation of the number of speakers and have a better speaker segmentation, the system scans for several thresholds until it finds an optimum on a hold-out dataset. We evaluated this approach using the Albayzin2018 dev2 dataset.

7. Experiments

In this section, we describe some experiments that give us a clue of the overall performance of our system. We evaluated our systems with Diarization Error Rate (DER), which is the most common metric for speaker diarization. The diarization error can be decomposed into speaker error, false alarm speech, missed speech and overlap speaker. Our DER tolerated errors within 250ms of a speaker transition and only scored the non-overlapping part of the segments because our model outputs single label for each frame. Our systems were evaluated on the Albayzin2018 dev2.

We employed the Albayzin2018 dev2 set as the initial part of our experiments. This set was divided into two parts, and we tune the parameters on one part and compute the DER performance on the other, which was similar to the Kaldi Callhome diarization recipe [3].

The DER results of different systems for the open and closed condition are shown in Table 1 and Table 2 respectively.

For the open condition, we examined four types of embeddings: i-vector. We explored two types of DNN based embedding architectures. The first one, the default Kaldi recipe for Voxceleb, is a TDNN for x-vector-basic [5, 6]. In this approach, each MFCC frame is passed through a sequence of TDNN layers. Then, a pooling layer accounts for the utterance level process and computes the mean and standard deviation of the TDNN output over time in a pooling layer. This intermediate representation, known as embedding, is projected to a lower dimension (512 in this case). The DNN output are the posterior probabilities of the training speakers. The objective function is cross entropy. We employed data set 3 for training the TDNN. The augmentation is performed as described in [7] using MUSAN noises.

For the second x-vector approach the pre-pooling layers are changed to factorized TDNNs (TDNN-F) with skip connections [8]. This new architecture reduces the number of parameters in the network by factorizing the weight matrix of each TDNN layer into the product of two low-rank matrices. The first factor is forced to be semi-orthogonal that will prevent the lost of information when projecting from high to low dimension. As in other architectures, skip connections are an option for this TDNN-F. Some input layers receive as input the output of the previous layer and other prior layers. The best solution so far is to have skip connection between low-rank interior layers in the TDNN-F. The x-vectors are of dimension 600.

As shown in Table 1, x-vector based systems outperform the i-vector based ones, which is consistent with previous studies. Among the four systems, the TDNN-F based x-vector performs the best. It outperforms the basic x-vector and i-vector by 1.27% and 3.90% absolute. Equal weighted score fusion further reduces the DER to 9.30%. It is interesting that the DER performance of x-vector based systems degrades when clustering with the actual number of speakers while it improves for the i-vector based systems. This indicates that i-vector based systems require more prior knowledge of the number of speakers.

Table 2 shows the DER results for the closed condition. The x-vector system achieves a DER of 24.03%, which is further improved to 22.26% if clustering with the oracle number of speakers. However, the performance of the x-vector is not as good as i-vector. We believe the reason is that we cannot obtain enough data to train a discriminative neural network. Even after data augmentation with the music, noise and speech we extracted from Albayzin2016 dataset, the training set only contained 332 hours of speech which was much smaller than the usual amount of data to train the x-vector system. Besides, since the recordings were from TV programs, a large number of speakers didn’t have enough corpus. The score fusion didn’t improve the system performance for the closed condition.
From our experiment, we also observe that the SAD is of vital importance. Since we don’t know the oracle SAD marks, the quality of the SAD is directly associated with the DER performance. Three different SAD models were evaluated, among which the 5-layer TDNN model trained on in-domain data performs the best. It outperforms the TDNN model trained on the Librispeech with same network architecture by 2.27% absolute. The energy based SAD is simple but the performance is worse than the TDNN models by a large margin. In our final system, we use the TDNN SAD trained on Albayzin2016 for both the open and closed condition.

Table 2: DER (%) comparison of different systems for the closed condition

<table>
<thead>
<tr>
<th>system</th>
<th>supervised calibration</th>
<th>oracle calibration</th>
<th>more than 10s</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-vector</td>
<td>13.39</td>
<td>13.46</td>
<td>13.68</td>
</tr>
<tr>
<td>i-vector</td>
<td>16.02</td>
<td>15.69</td>
<td>15.08</td>
</tr>
<tr>
<td>BNF-i-vector</td>
<td>16.40</td>
<td>15.88</td>
<td>14.83</td>
</tr>
<tr>
<td>fusion</td>
<td><strong>9.39</strong></td>
<td>10.83</td>
<td>11.87</td>
</tr>
</tbody>
</table>

Table 3: DER (%) of basic x-vector system with different SAD

<table>
<thead>
<tr>
<th>SAD</th>
<th>DER</th>
</tr>
</thead>
<tbody>
<tr>
<td>energy based SAD</td>
<td>21.52</td>
</tr>
<tr>
<td>TDNN SAD trained on Librispeech</td>
<td>15.66</td>
</tr>
<tr>
<td>TDNN SAD trained on Albayzin2016</td>
<td><strong>13.39</strong></td>
</tr>
</tbody>
</table>

8. Future Work

Although we largely reduce the diarization error with in-domain SAD and system fusion, there are still many problems to investigate. The first is the overlap problem. Our current system cannot handle the overlapping speech, since it predicts a single speaker label for each frame. However, solving this problem is not easy. The whole procedure of the diarization might change to predict multiple labels for one frame. Second, as discussed in the former part, the number of speakers estimated by the supervised calibration is not very close to the actual number. Besides, clustering with the oracle number of speakers sometimes even degrades the system. Whether there exists better methods to control the clustering process, especially for the condition with many speakers, requires further studies. Third, due to the time limit, we didn’t include the re-segmentation process in our system. We will add this part later to see if it can further boost the system performance.

9. Conclusions

This is the submission for the JHU Diarization system. We tried our out-of-the-box system in this new scenario that contained broadcast news in a new language. Two main solutions were proposed for the closed and the open conditions: i-vector and x-vector. I-vector was showed to be best suited for the closed condition, due to the small amount of training data. For the open condition, the best results were obtained by the x-vector based system. However, having a score fusion, before the clustering gave noticeable improvements. We are still planning to do some re-segmentation in future versions.

10. Acknowledgements

The authors would like to thank David Snyder for his help in this project.

11. References
GTM-IRLab Systems for Albayzin 2018 Search on Speech Evaluation

Paula Lopez-Otero1, Laura Docio-Fernandez2

1 Universidade da Coruña - CITIC, Information Retrieval Lab
2 Universidade de Vigo - atlanTTic Research Center, Multimedia Technology Group

paula.lopez.oteroudc.gal, ldocio@cts.uvigo.es

Abstract

This paper describes the systems developed by the GTM-IRLab team for the Albayzin 2018 Search on Speech evaluation. The system for the spoken term detection task consists in the fusion of two subsystems: a large vocabulary continuous speech recognition strategy that uses the proxy words approach for out-of-vocabulary terms, and a phonetic search system based on the probabilistic retrieval model for information retrieval. The query-by-example spoken term detection system is the result of fusing four subsystems: three of them are based on dynamic time warping search using different representations of the waveforms, namely Gaussian posteriorgrams, phoneme posteriorgrams and a large set of low-level descriptors; and the other one is the phonetic search system used for spoken term detection with some modifications to manage spoken queries.

Index Terms: Spoken term detection, query-by-example spoken term detection, large vocabulary continuous speech recognition, out-of-vocabulary terms, phoneme posteriorgrams, Gaussian posteriorgrams, probabilistic information retrieval, phonetic search

1. Introduction

In this paper, the systems developed by the GTM-IRLab team for the Albayzin 2018 Search on Speech evaluation are described. Systems were submitted to the spoken term detection (STD) and the query-by-example spoken term detection (QbESTD) tasks.

In the STD task, a fusion of two subsystems was proposed. The first system consists in a strategy based on large vocabulary continuous speech recognition (LVCSR). This LVCSR system was built using the Kaldi toolkit [1] to train a set of acoustic models, to generate the output lattices and to perform lattice indexing and term search [2]. The proxy words strategy described in [3] was used to deal with out-of-vocabulary (OOV) terms. The second system performs phonetic search (PS) using an approach that adapts the probabilistic retrieval model [4] for information retrieval to the search on speech task similarly as described in [5, 6].

For the QbESTD task, the proposed system consists in a fusion of four different systems. Three of them rely on dynamic time warping (DTW) search with different representations of the speech data, namely phoneme posteriorgrams [7], low-level descriptors [8] and Gaussian posteriorgrams [9]. The fourth system is an adaptation of the PS approach used for STD that copes with spoken queries.

The rest of this paper is organized as follows: Section 2 and 3 describe the systems for the STD and QbESTD tasks, respectively; Section 4 presents the preliminary results obtained for the different tasks on the development data; and Section 5 presents some conclusions extracted from the experimental validation of the different systems.

2. Spoken term detection system

The proposed STD system consists in the fusion of two subsystems: one is based in an LVCSR system while the other is a PS system that adapts the probabilistic retrieval model for information retrieval to the STD task.

2.1. LVCSR system

An LVCSR system was built using the Kaldi open-source toolkit [1]. Deep neural network (DNN) based acoustic models were used; specifically, a DNN-based context-dependent speech recognizer was trained following Karel Veselý’s DNN training approach [10]. The input acoustic features to the neural network are 40 dimensional Mel-frequency cepstral coefficients (MFCCs) augmented with three pitch and voicing related features [11], and appended with their delta and acceleration coefficients. The DNN has 6 hidden layers with 2048 neurons each. Each speech frame is spliced across ±5 frames to produce 1419 dimensional vectors which are the input to the first layer, and the output layer is a soft-max layer representing the log-posteriors of the context-dependent HMM states.

The Kaldi LVCSR decoder generates word lattices [12] using the above DNN-based acoustic models. These lattices are processed using the lattice indexing technique described in [2] so that the lattices of all the utterances in the search collection are converted from individual weighted finite state transducers (WFST) to a single generalized factor transducer structure in which the start-time, end-time and lattice posterior probability of each word token is stored as a 3-dimensional cost. This factor transducer is actually an inverted index of all word sequences seen in the lattices. Thus, given a list of keywords or phrases, a simple finite state machine is created such that it accepts the keywords/phrases and composes them with the factor transducer to obtain all the occurrences of the keywords/phrases in the search collection.

The proxy words strategy included in Kaldi [3] was used for OOV term detection. This approach uses phone confusion to find the in-vocabulary (INV) term that is the most similar, in terms of its phonetic content, to the corresponding OOV term, and search is performed using the INV term.

The data used to train the acoustic models of this LVCSR system were extracted from the Spanish material used in the 2006 TC-STAR automatic speech recognition evaluation campaign1 and from the Galician broadcast news database Transcrigal [13]. It must be noted that all the non-speech parts as well as the speech parts corresponding to transcriptions with pronunciation errors, incomplete sentences and short speech utterances were discarded, so in the end the acoustic training material consisted of approximately 104 hours and 30 minutes.

The language model (LM) was constructed using a text database of 150 MWords composed of material from several

http://www.tc-star.org
sources (transcriptions of European and Spanish Parliaments from the TC-STAR database, subtitles, books, newspapers, online courses and transcriptions of the Mavir sessions included in the development set) [14]. Specifically, two fourgram-based language models were trained following the Kneser-Ney discounting strategy using the SRILM toolkit [15], and the final LM was obtained by mixing both LMs using the SRILM static n-gram interpolation functionality. One of the LMs was trained using the RTVE2018 subtitles data provided for the Albayzin 2018 Text-to-Speech challenge and the other LM was built using the other text corpora. The LM vocabulary size was limited to the most frequent 300K words and, for each search task, the set of OOV keywords were removed from the language model.

2.2. PS system

A system based on phonetic search following the probabilistic retrieval model for information retrieval was developed for the STD task:

- Indexing. First, the phone transcription of each document is obtained, and then the documents are indexed in terms of phone n-grams of different size [16, 5]. According to the probabilistic retrieval model, each document is represented by means of a language model [4]. In this case, given that the phone transcriptions have errors, several hypotheses for the best transcription are used to improve the quality of the language model [6].

- Search. First, a phonetic transcription of the query is obtained using the grapheme-to-phoneme model for Spanish included in Cotovia [17]. Then, the query is searched within the different indices, and a score for each document is computed following the query likelihood retrieval model [18]. It must be noted that this model sorts the documents according to how likely they contain the query, but the start and end times of the match are required in this task. To obtain these times, the phone transcription of the query is aligned to that of the document by computing their minimum edit distance, and this allows the recovery of the start and end times since they are stored in the index. In addition, the minimum edit distance is used to penalize the score returned by the query likelihood retrieval model as described in [6].

The minimum and maximum size of the n-grams were set to 1 and 5, respectively, according to [5]. The different hypotheses for the phone transcriptions of the documents were extracted from the phone lattice obtained employing the LVCSR system described above, and the number of hypotheses to be used for indexing was empirically set to 40. Indexing and search were performed using Lucene.

2.3. Fusion

Discriminative calibration and fusion were applied in order to combine the outputs of the different STD systems [19]. The global minimum score produced by the system for all queries was used to hypothesize the missing scores. After normalization, calibration and fusion parameters were estimated by logistic regression on a development dataset in order to obtain improved discriminative and well-calibrated scores [20]. Calibration and fusion training was performed using the Bosaris toolkit [21].

3. Query-by-example spoken term detection system

The primary system submitted for the QbESTD evaluation consists in the fusion of four systems. Three of those systems follow the same scheme: first, feature extraction is performed in order to represent the queries and documents by means of feature vectors; then, the queries are searched within the documents using a search approach based on DTW; finally, a score normalization step is performed. The other system is an adaptation of the PS system described above to the QbESTD task.

3.1. DTW-based systems

3.1.1. Speech representation

Three different approaches for speech representation were used; given a query Q with n frames (and equivalently, a document D with m frames), these representations result in a set Q = \{q_1, \ldots, q_n\} of n vectors of dimension U (and equivalently, a set D = \{d_1, \ldots, d_m\} of m vectors of dimension U):

- Phoneme posteriorgram (PhnPost). One subsystem relies on phoneme posteriorgrams [7] for speech representation: given a speech document and a phoneme recognizer with U phonetic units, the a posteriori probability of each phonetic unit is computed for each time frame, leading to a set of vectors of dimension U that represent the probability of each phonetic unit at every time instant. The English (EN) phone decoder developed by the Brno University of Technology was used to obtain phoneme posteriorgrams; in this decoder, each phonetic unit has three different states and a posterior probability is output for each of them, so they were combined in order to obtain one posterior probability for each unit [22]. After obtaining the posteriors, a Gaussian softening was applied in order to have Gaussian distributed probabilities [23].

- Low-level descriptors (LLD). A large set of features, summarised in Table 1, was used to represent the queries and documents: these features, obtained using the OpenSMILE feature extraction toolkit [24], were extracted every 10 ms using a 25 ms window, except for F0, probability of voicing, jitter, shimmer and HNR, for which a 60 ms window was used.

- Gaussian posteriorgram (GP). Gaussian posteriorgrams [9] were used to represent the audio documents and queries. Given a Gaussian mixture model (GMM) with U Gaussians, the a posteriori probability of each Gaussian is computed for each time frame, leading to a set of vectors of dimension U that represent the probability of each Gaussian at every time instant. In this system, 19 MFCCs were extracted from the waveforms, accompanied with their energy, delta and acceleration coefficients. Feature extraction and Gaussian posteriorgram computation were performed using the Kaldi toolkit [1]. The GMM was trained using MAVIR training and development data, as well as RTVE development recordings.
Table 1: Acoustic features used in the proposed search on speech system.

<table>
<thead>
<tr>
<th>Description</th>
<th># features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum of auditory spectra</td>
<td>1</td>
</tr>
<tr>
<td>Zero-crossing rate</td>
<td>1</td>
</tr>
<tr>
<td>Sum of RASTA style filtering auditory spectra</td>
<td>1</td>
</tr>
<tr>
<td>Frame intensity</td>
<td>1</td>
</tr>
<tr>
<td>Frame loudness</td>
<td>1</td>
</tr>
<tr>
<td>Root mean square energy and log-energy</td>
<td>2</td>
</tr>
<tr>
<td>Energy in frequency bands 250-650 Hz (energy 250-650) and 1000-4000 Hz</td>
<td>2</td>
</tr>
<tr>
<td>Spectral Rolloff points at 25%, 50%, 75%, 90%</td>
<td>4</td>
</tr>
<tr>
<td>Spectral flux</td>
<td>1</td>
</tr>
<tr>
<td>Spectral entropy</td>
<td>1</td>
</tr>
<tr>
<td>Spectral variance</td>
<td>1</td>
</tr>
<tr>
<td>Spectral skewness</td>
<td>1</td>
</tr>
<tr>
<td>Spectral kurtosis</td>
<td>1</td>
</tr>
<tr>
<td>Psychoacoustical sharpness</td>
<td>1</td>
</tr>
<tr>
<td>Spectral harmonicity</td>
<td>1</td>
</tr>
<tr>
<td>Spectral flatness</td>
<td>1</td>
</tr>
<tr>
<td>Mel-frequency cepstral coefficients</td>
<td>16</td>
</tr>
<tr>
<td>MFCC filterbank</td>
<td>26</td>
</tr>
<tr>
<td>Line spectral pairs</td>
<td>8</td>
</tr>
<tr>
<td>Cepstral perceptual linear predictive coefficients</td>
<td>9</td>
</tr>
<tr>
<td>RASTA PLP coefficients</td>
<td>9</td>
</tr>
<tr>
<td>Fundamental frequency (F0)</td>
<td>1</td>
</tr>
<tr>
<td>Probability of voicing</td>
<td>1</td>
</tr>
<tr>
<td>Jitter</td>
<td>2</td>
</tr>
<tr>
<td>Shimmer</td>
<td>1</td>
</tr>
<tr>
<td>Log harmonics-to-noise ratio (logHNR)</td>
<td>1</td>
</tr>
<tr>
<td>LCP formant frequencies and bandwidths</td>
<td>6</td>
</tr>
<tr>
<td>Formant frame intensity</td>
<td>1</td>
</tr>
<tr>
<td>Delta</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>208</td>
</tr>
</tbody>
</table>

3.1.2. Search algorithm

The search stage was carried out using the subsequence DTW (S-DTW) [25] variant of the classical DTW approach. To perform S-DTW, first a cost matrix \( M(n \times m) \) must be defined, in which the rows and columns correspond to the query and document frames, respectively:

\[
M_{i,j} = \begin{cases} 
  c(q_i, d_j) & \text{if } i = 0 \\
  c(q_i, d_j) + M_{i-1,0} & \text{if } i > 0, j = 0 \\
  c(q_i, d_j) + M^*(i, j) & \text{else}
\end{cases}
\]

where \( c(q_i, d_j) \) is a function that defines the cost between the query vector \( q_i \) and the document vector \( d_j \), and

\[
M^*(i, j) = \min \{ M_{i-1,j}, M_{i-1,j-1}, M_{i,j-1} \}
\]

Pearson’s correlation coefficient \( r \) [26] was the metric used to define the cost function by mapping it into the interval \([0,1]\) applying the following transformation:

\[
c(q_i, d_j) = \frac{1 - r(q_i, d_j)}{2}
\]

Once matrix \( M \) is computed, the end of the best warping path between \( Q \) and \( D \) is obtained as

\[
b^* = \arg \min_{b \in 1, \ldots, m} M(n, b)
\]

The starting point of the path ending at \( b^* \), namely \( a^* \), is computed by backtracking, hence obtaining the best warping path \( P(Q, D) = \{p_1, \ldots, p_k, \ldots, p_K\} \), where \( p_k = (i_k, j_k) \). i.e. the \( k \)-th element of the path is formed by \( q_{i_k} \) and \( d_{j_k} \), and \( K \) is the length of the warping path.

It is possible that a query \( Q \) appears several times in a document \( D \), especially if \( D \) is a long recording. Hence, not only the best warping path must be detected but also others that are less likely. One approach to overcome this issue consists in detecting a given number of candidate matches \( n_c \); every time a warping path, that ends at frame \( b^* \), is detected, \( M(n, b^*) \) is set to \( \infty \) in order to ignore this element in the future.

A score must be assigned to every detection of a query \( Q \) in a document \( D \). First, the cumulative cost of the warping path \( M_{n,b^*} \) is length-normalized [27] and, after that, z-norm is applied so that all the scores of all the queries have the same distribution [28].

3.2. PS system

The system described in Section 2.2 was also used for QbESTD. Since, in this experimental setup, the queries are spoken, the LVCSR system described in Section 2.1 was used to obtain phone transcriptions of the queries. In this system, the number of transcription hypotheses of the documents was empirically set to 50.

3.3. Fusion

The fusion strategy described in Section 2.3 was used to combine the QbESTD systems described in this section.

4. Preliminary Results

The systems described in the previous sections were evaluated in terms of the average term weighted value (ATWV) and maximum term weighted value (MTWV), which are the evaluation metrics defined for Albayzin 2018 evaluation. The results included in this section were achieved using the development data provided by the organizers. Since two different datasets (MAVIR and RTVE) were used for development, and in order to avoid overfitting when choosing the decision threshold, the groundtruth labels of MAVIR and RTVE were joined into a single set (namely MAVIR+RTVE) to compute the decision
Table 2: STD results on development data

<table>
<thead>
<tr>
<th>System</th>
<th>MAVIR</th>
<th>RTVE</th>
<th>MAVIR+RTVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVCSR (con1)</td>
<td>0.5314</td>
<td>0.5179</td>
<td>0.5976</td>
</tr>
<tr>
<td>PS (con2)</td>
<td>0.4828</td>
<td>0.4739</td>
<td>0.6286</td>
</tr>
<tr>
<td>LVCSR-NP (con3)</td>
<td>0.5068</td>
<td>0.4079</td>
<td>0.5801</td>
</tr>
<tr>
<td>Fusion (pri)</td>
<td>0.5470</td>
<td>0.5290</td>
<td>0.6550</td>
</tr>
</tbody>
</table>

Table 3: QbESTD results on development data

<table>
<thead>
<tr>
<th>System</th>
<th>MAVIR</th>
<th>RTVE</th>
<th>MAVIR+RTVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PhnPost (con2)</td>
<td>0.1971</td>
<td>0.1742</td>
<td>0.7145</td>
</tr>
<tr>
<td>LLD</td>
<td>0.2017</td>
<td>0.1774</td>
<td>0.7136</td>
</tr>
<tr>
<td>GP</td>
<td>0.1877</td>
<td>0.1628</td>
<td>0.6731</td>
</tr>
<tr>
<td>PS (con3)</td>
<td>0.2383</td>
<td>0.2029</td>
<td>0.3540</td>
</tr>
<tr>
<td>Fusion DTW (con1)</td>
<td>0.2699</td>
<td>0.2649</td>
<td>0.7211</td>
</tr>
<tr>
<td>Fusion (pri)</td>
<td>0.2896</td>
<td>0.2470</td>
<td>0.7273</td>
</tr>
</tbody>
</table>

threshold, which was subsequently applied to each dataset individually.

4.1. STD experiments

Table 2 shows the results achieved using the systems described in Section 2. The table also includes an additional system, namely LVCSR-NP, which consists in the aforementioned LVCSR without using the proxy words strategy for OOV terms; this means that the LVCSR-NP system does not detect any OOV terms. Comparing LVCSR-NP and LVCSR systems, it can be seen that using the proxy words strategy is beneficial specially when dealing with MAVIR data. The table also shows that the PS system outperforms the LVCSR system on RTVE dataset, and it also leads to a better overall result. The combination of both systems achieves a significant improvement in all the experimental conditions, which suggests that both strategies are strongly complementary.

4.2. QbESTD experiments

Table 3 shows the results achieved by the QbESTD systems described in Section 3. The best performance in MAVIR data was achieved with the PS system, which also exhibited the lowest performance in RTVE data. PhnPost and LLD systems achieved almost the same results for RTVE and MAVIR+RTVE data.

The table also displays the results obtained when fusing the three DTW approaches (Fusion DTW) and when fusing the four systems (Fusion). The MTWV is always higher when fusing the four systems but, for the individual datasets, the ATWV is higher when fusing only the DTW systems. Nevertheless, the overall result is better when combining the four systems, so this system was selected as the primary (pri) for this evaluation, while the fusion of the three DTW systems was presented as contrastive (con1).

5. Conclusions and future work

This paper presented the systems developed for the STD and QbESTD tasks of Albayzin 2018 Search on Speech evaluation. The STD system consists in a fusion of a LVCSR system with a phonetic search system based on the probabilistic retrieval model for information retrieval. The LVCSR system relied on the proxy words approach for OOV words, which were also managed by the phonetic search system. The QbESTD system is a fusion of three DTW-based systems with the phonetic search system used in the STD task.

The performance obtained in STD and QbESTD tasks are not straightforwardly comparable because the queries used to compute the evaluation metrics are not the same for both tasks, but the results suggest that spoken queries lead to better results in RTVE dataset. This might be caused by a greater amount of OOV words, so this will be investigated by further analysis of the results.

In future work, a system that combines word-level and phone-level representations with the probabilistic retrieval model for information retrieval will be assessed. This idea is motivated by the fact that, according to the results exhibited in the STD task, the LVCSR and phonetic search systems are strongly complementary, and designing smart combination strategies might improve the performance of logistic regression fusion.

The DTW-based systems for QbESTD used in this paper are language-independent, i.e. the system can be used regardless of the language spoken in the recordings. Given that a LVCSR system for Spanish was trained for the STD system, the use of the activations of the LVCSR network will be investigated in future work in order to assess QbESTD performance in a language-dependent setting.

6. Acknowledgements

This work has received financial support from i) “Ministerio de Economía y Competitividad” of the Government of Spain and the European Regional Development Fund (ERDF) under the research projects TIN2015-64282-R and TEC2015-65345-P, ii) Xunta de Galicia (projects GPC ED431B 2016/035 and GRC 2014/024), and iii) Xunta de Galicia – “Consellería de Cultura, Educación e Ordenación Universitaria” and the ERDF through the 2016-2019 accreditations ED431G01 (“Centro singular de investigación de Galicia”) and ED431G04 (“Agrupación estratéxica consolidada”).

7. References


AUDIAS-CEU: A Language-independent approach for the Query-by-Example Spoken Term Detection task of the Search on Speech ALBAYZIN 2018 evaluation

Maria Cabello¹, Doroteo T. Toledano¹, Javier Tejedor²

¹AUDIAS, Universidad Autónoma de Madrid
²Universidad San Pablo-CEU, CEU Universities
m.cabello.a@gmail.com, doroteo.torre@uam.es, javier.tejedornoquerales@ceu.es

Abstract

Query-by-Example Spoken Term Detection is the task of detecting query occurrences within speech data (henceforth utterances). Our submission is based on a language-independent template matching approach. First, queries and utterances are represented as phonetic posteriorgrams computed for English language with the phoneme decoder developed by the Brno University of Technology. Next, the Subsequence Dynamic Time Warping algorithm with a modified Pearson correlation coefficient as cost measure is employed to hypothesize detections. Results on development data showed an ATWV=0.1774 with MAVR data and an ATWV=0.0365 with RTVE data.

Index Terms: language-independent QbE STD, template matching, SDTW

1. Introduction

The large amount of heterogeneous speech data stored in audio and audiovisual repositories makes it necessary to develop efficient methods for speech information retrieval. There are different speech information retrieval tasks, including spoken document retrieval (SDR), keyword spotting (KWS), spoken term detection (STD), and query-by-example spoken term detection (QbE STD). The advance of the technology in these tasks has been evaluated through different international evaluations related to SDR, STD, and QbE STD [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] for different languages (English, Arabic, Mandarin, Spanish, Japanese, and low-resource language such as Swahili, Tamil, and Vietnamese). Specifically, Spanish language has been employed with the STD/QbE STD ALBAYZIN evaluations held in 2012, 2014 and 2016 [7, 8, 9, 10, 11].

Query-by-example Spoken Term Detection aims to retrieve data from a speech repository (henceforth utterance) given an acoustic query containing the term of interest as input. QbE STD has been mainly addressed from three different approaches: methods based on the word/subword transcription of the query that typically employ a word/phone-based speech recognition system for query detection [12, 13], methods based on template matching of features that are typically based on posteriorgram-based units and DTW-like search for query detections [14, 15, 16, 17], and hybrid approaches that take advantage of both approaches [18, 19, 20, 21].

This paper presents the system submitted by the AUDIAS-CEU research team to the QbE STD task of the Search on Speech ALBAYZIN 2018 evaluation [22], which deals with the Spanish language. However, our submission does not employ the target language, and hence is based on a language-independent approach that can be used for any target language. First, phoneme posteriorgrams are computed for query/utterance representation. These phoneme posteriorgrams are computed from neural networks trained for the English language. Next, the Subsequence Dynamic Time Warping (SDTW) algorithm generates the query occurrences. Our system is largely based on the winner system of the QbE STD task of the Search on Speech ALBAYZIN 2016 evaluation [23].

The rest of the paper is organized as follows: Section 2 presents the system submitted to the evaluation, Section 3 presents the experiments and the results obtained in the development data provided by the organizers, and Section 4 concludes the paper.

2. QbE STD system

The QbE STD system, whose architecture is presented in Figure 1, integrates two different stages: feature extraction and query detection, which are explained next.

![QbE STD system architecture.](image)

2.1. Feature extraction

The English phoneme recognizer development by the Brno University of Technology [24] has been employed to compute 3-state phoneme posteriorgrams that represent both the queries and the utterances. This phoneme recognizer contains 39 units, which correspond to the 39 phonemes in English plus a non-speech unit to represent some other phenomena in speech such as laugh, noise, short silences, etc. These phoneme posteriorgrams have been computed each 10ms of speech, which have been further processed to compute a single posterior probability per phoneme. This posterior probability is obtained by summing the posterior probabilities of the three states corresponding to the given phoneme.

2.2. Query detection

Query detection involves different stages, as presented in Figure 2.

First, a cost matrix that stores the similarity between every query/utterance pair is computed. The Pearson correlation coefficient [25] has been employed to build the cost matrix, as presented in Equation 1:
\[
r(x_n, y_m) = \frac{U(x_n \cdot y_m) - ||x_m|| ||y_m|| \sqrt{||x_n||^2 - ||x_n||^2}(U ||y_m||^2 - ||y_m||^2)}{U ||x_n||^2 - ||x_n||^2}, \tag{1}
\]

where \( x_n \) represents the query phoneme posterigrams, \( y_m \) represents the utterance phoneme posterigrams, and \( U \) represents the number of phoneme units (40 in our case).

Then, the Pearson correlation coefficient is mapped into the interval [0 1], as given in Equation 2:

\[
c(x_n, y_m) = \frac{1 - r(x_n, y_m)}{2}, \tag{2}
\]

where \( c(x_n, y_m) \) represents the cost matrix used during the S-DTW search.

Therefore, the cost \( c(x_n, y_m) \) can take the values of 1 (when \( r = -1 \)), 0.5 (when \( r = 0 \)), or 0 (when \( r = 1 \)). Figure 3 represents the cost matrix example with the standard Pearson correlation coefficient computation.

The final cost used during the search has been modified as follows: When \( r < 0 \), \( r \) has been assigned the value of 0. Next, \( c(x_n, y_m) = 1 - r(x_n, y_m) \). Therefore, for all the Pearson correlation coefficient values lower or equal to 0, the cost will be maximum, hence promoting the differences between aligned and non-aligned sequences in the next stage.

Figure 4 shows the cost matrix example with this modification of the Pearson correlation coefficient computation. This modification leads to more differences in the costs between query and utterance frames.

\[
D_{n,m} = \begin{cases} 
c(x_n, y_m) & \text{if } n = 0 \\
c(x_n, y_m) + D_{n-1,0} & \text{if } n > 0, m = 0 \\
c(x_n, y_m) + D'(n, m) & \text{else,}
\end{cases} \tag{3}
\]

where

\[
D'(n, m) = \min (D_{n-1,m}, D_{n-1,m-1}, D_{n,m-1}) \tag{4}
\]

which implies that only horizontal, vertical, and diagonal path movements are allowed in the search.
Figure 5 shows the accumulated cost matrix from the cost matrix presented in Figure 3 (i.e., with the standard Pearson correlation coefficient computation), and Figure 6 shows that of the cost matrix presented in Figure 4 (i.e., with the modified Pearson correlation coefficient). The accumulated cost matrix from the modified Pearson correlation coefficient shows more cost in non-occurrence regions, which favors the final query detection.

![Figure 5](image5.png)

**Figure 5:** Accumulated cost matrix from Figure 3 with the standard Pearson correlation coefficient.

![Figure 6](image6.png)

**Figure 6:** Accumulated cost matrix from Figure 4 with the modified Pearson correlation coefficient.

To hypothesize detections, a distance function $\Delta$ needs to be defined. This value corresponds to the last row/column of the accumulated cost matrix $D$, and is used as the initial value from which the minimum values that suggest possible paths are computed. Then, the S-DTW computes a global minimum $b_{\min}$ (also referred as $b^*$ in Figure 2) in the accumulated cost matrix from which all the possible query detections are considered. This $b_{\min}$ has to be lower than a predefined threshold $\tau$ tuned on MAVIR development data. Next, the S-DTW computes all the local minima $b^*$ that appear in the accumulated cost matrix. These local minima $b^*$ need to be lower than a second threshold $\tau_2 = \tau \times b_{\min}$ to be considered as optimal paths where query detections reside. This second threshold $\tau_2$ has been also tuned on MAVIR development data.

For each of values of $b^*$ that meets the afore-mentioned conditions, the optimal paths $(a^*, b^*)$ that represent the query detections are found as follows: Let $p = (p_1, \ldots, p_l)$ be a possible optimal path. Starting at $p_1 = b^*$, a reverse path that ends at $n = 1$ (i.e., the first frame of the query) is computed as presented in Equation 5:

$$p_{l-1} = \text{argmin}(D(n-1, m-1), D(n-1, m), D(n, m-1)).$$

Finally, a neighbourhood search is carried out so that all the paths (i.e., query detections) which overlap 500 ms from a previously obtained optimal path are rejected in the final system output. An example of an optimal path found is presented in Figure 7.

### 3. Experiments and results

Experiments are carried out on the development data provided by the organizers for the QbE STD task. Two different databases were experimented with: MAVIR database, which comprises a set of talks extracted from the Spanish MAVIR workshops held in 2006, 2007, and 2008 (Corpus MAVIR 2006, 2007, and 2008) corresponding to Spanish language, and RTVE database, which comprises different Radio Televisión Española (RTVE) programs recorded from 2015 to 2018. For the MAVIR database, about 1 hour of speech material in total, extracted from 2 audio files was provided by the organizers, in which 102 queries extracted from the same development data were searched. For the RTVE database, about 15 hours of speech, extracted from 12 audio files, were provided by the organizers. In these RTVE data, 103 queries were searched. Organizers also provided with training data. However, since our submission is based on a previously trained phoneme recognizer for English language, the system does not employ any training data.

Results are shown in Table 1 for MAVIR and RTVE development data. These results show moderate performance for MAVIR data and a worse performance on RTVE data. This worse performance on RTVE data may be due to the optimal parameters found on MAVIR data were employed for the RTVE data, and no additional tuning on these data were carried out. Better performance should be obtained in case RTVE data were also fine-tuned.

### 4. Conclusions

This paper presents the AUDIAR-CEU submission for the QbE STD task of the Search on Speech ALBAYZIN 2018 evaluation. The system relies on a language-independent approach for QbE STD, since no prior information of the Spanish language is employed for system building. Phoneme posteriorgrams and Subsequence Dynamic Time Warping with a modified Pearson correlation coefficient as cost measure were employed for system construction. System design was largely based on the winner system of the QbE STD task of the Search on Speech ALBAYZIN 2016 evaluation [23].

Future work will include some fusion techniques to get advantage of the query detections from different phoneme decoders, and feature selection techniques to retain the most meaningful phoneme units for each language.

---

1http://www.mavir.net

<table>
<thead>
<tr>
<th>Database</th>
<th>MTWV</th>
<th>ATWV</th>
<th>p(FA)</th>
<th>p(Miss)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAVIR</td>
<td>0.1823</td>
<td>0.1774</td>
<td>0.00002</td>
<td>0.801</td>
</tr>
<tr>
<td>RTVE</td>
<td>0.0365</td>
<td>0.0365</td>
<td>0.00000</td>
<td>0.963</td>
</tr>
</tbody>
</table>
5. Acknowledgements

This work was partially supported by the project “DSSL: Redes Profundas y Modelos de Subespacios para Detec-cin y Seguimiento de Lector, Idioma y Enfermedades De-generativas a partir de la Voz” (TEC2015-68172-C2-1-P, MINECO/FEDER).

6. References


GTTS-EHU Systems for the Albayzin 2018 Search on Speech Evaluation

Luis Javier Rodríguez-Fuentes, Mikel Peñagarikano, Aparo Varona, Germán Bordel

Departamento de Electricidad y Electrónica UPV/EHU
Barrio Sarriena s/n, 48940 Leioa, Bizkaia, Spain
luisjavier.rodriguez@ehu.eus

Abstract

This paper describes the systems developed by GTTS-EHU for the QbE-STD and STD tasks of the Albayzin 2018 Search on Speech Evaluation. Stacked bottleneck features (sBNF) are used as frame-level acoustic representation for both audio documents and spoken queries. In QbE-STD, a flavour of segmental DTW (originally developed for MediaEval 2013) is used to perform the search, which iteratively finds the match that minimizes the average distance between two test-normalized sBNF vectors, until either a maximum number of hits is obtained or the score does not attain a given threshold. The STD task is performed by synthesizing spoken queries (using publicly available TTS APIs), then averaging their sBNF representations and using the average query for QbE-STD. A publicly available toolkit developed by BUT/Phonexia has been used to extract three sBNF sets, trained for English monophone and triphone state posteriors (contrastive systems 3 and 4) and for multilingual triphone posteriors (contrastive system 2), respectively. The concatenation of the three sBNF sets has been also tested (contrastive system 1). The primary system consists of a discriminative fusion of the four contrastive systems. Detection scores are normalized on a query-by-query basis (qnorm), calibrated and, if two or more systems are considered, fused with other scores. Calibration and fusion parameters are discriminatively estimated using the ground truth of development data. Finally, due to a lack of robustness in calibration, Yes/No decisions are made by applying the MTWV thresholds obtained for the development sets, except for the COREMAH test set. In this case, calibration is based on the MAVIR corpus, and the 15% highest scores are taken as positive (Yes) detections.

Index Terms: Spoken Term Detection, Query-by-Example
Spoken Term Detection, Bottleneck features, Dynamic Time Warping

1. Introduction

The main goal of the participation of GTTS in the Albayzin 2018 Search on Speech Evaluation was to upgrade the QBE-STD systems that we developed for MediaEval 2013 and 2014 [1][2] by: (1) using an external VAD module; (2) replacing phonetic posterior by bottleneck features as frame-level features (which might imply changing some aspects of the methodology); and (3) handling the case of no development data by applying cross-condition calibration and heuristic thresholding. Also, as a proof-of-concept attempt to overcome the issue of OOV words, we have developed STD systems by applying public TTS APIs to synthesize a number of spoken instances of each term, then computing an average query (following the approach developed for MediaEval 2013 [3]) and using it to perform QBE-STD.

QBE-STD results on development data reveal that calibration models are not well estimated, probably due to a lack of detections for target trials. In particular, the threshold given by the application model parameters ($P_{\text{target}} = 0.0001$, $C_{\text{miss}} = 1$ and $C_{fa} = 0.1$) is too high compared to the MTWV threshold.

In the case of STD, the lack of detections is even more remarkable, with $P_{\text{miss}} \approx 0.8$ for extremely low thresholds. This could be due to an acoustic mismatch between the synthesized queries and the test audio signals, which might be blocking the DTW-based search, since the score yielded by the best match would fall below the established threshold. The low amount of detections (especially, for target trials) not only yields high miss error rates but also makes the calibration model to be poorly estimated. This may explain why the MTWV threshold for STD scores on development data is much lower that that provided by the application model.

In both cases (QBE-STD and STD), the parameters of the DTW-based search should be further tuned in order to get a larger amount of target and non-target detections.

2. QBE-STD systems

In the design of our previous QBE-STD systems, the frontend exploited existing software (e.g. the BUT phone decoders for Czech, Hungarian and Russian [4]), whereas the backend (search, calibration and fusion) was almost entirely developed by our group (and collaborators) [3][5]. For this evaluation, we have applied an external VAD module and extracted a new set of frame-level features.

2.1. Voice Activity Detection (VAD)

In our previous QBE-STD systems, VAD was performed by using the posteriori provided BUT phone decoders: first, the posteriori of non-phonetic units were added; then, if this aggregated non-phonetic posterior was higher than any other (phonetic) posterior, the frame was labeled as non-speech; otherwise it was labeled as speech. In this evaluation, we are not using posteriori anymore, so we cannot apply the same procedure. Instead, we apply the Python interface to a VAD module developed by Google for the WebRTC project [6], based on Gaussian distributions of speech and non-speech features. Given an audio file, our VAD module produces two output files: the first one (required by the feature extraction module) is an HTK .lab file specifying speech and non-speech segments, whereas the second one (used by the search procedure) is a text file (.txt) containing a sequence of 1’s and 0’s (one per line) indicating speech/non-speech frames.

2.2. Bottleneck features

We have updated the feature extraction module of our previous QBE-STD systems with the stacked bottleneck features (sBNF) recently presented by BUT/Phonexia [7]. Actually, three different neural networks are applied, each one trained to classify a different set of acoustic units and later optimized to language
recognition tasks. The first network was trained on telephone speech (8 kHz) from the English Fisher corpus [8] with 120 monophone state targets (FisherMono); the second one was also trained on the Fisher corpus but with 2423 triphone tied-state targets (FisherTri); the third network was trained on telephone speech (8 kHz) in 17 languages taken from the IARPA Babel program [9], with 3096 stacked monophone state targets for the 17 languages involved (BabelMulti).

The architecture of these networks consists of two stages. The first one is a standard bottleneck network fed with low-level acoustic features spanning 10 frames (100 ms), the bottleneck size being 80. The second stage takes as input five equally spaced BNFs of the first stage, spanning 31 frames (310 ms), and is trained on the same targets as the first stage, with the same bottleneck size (80). The bottleneck features extracted from the second stage are known as stacked bottleneck features (sBNF). Alternatively, instead of sBNF, the extractor can output target posteriors.

The operation of BUT/Phonexia sBNF extractors requires an external VAD module providing speech/non-speech information through an HTK .lab file. If no external VAD is provided, a simple energy-based VAD is computed internally. In this evaluation, we have applied the WebRTC VAD described above.

Our first aim was to replace old BUT by new BUT/Phonexia posteriors, but the huge size of FisherTri (2423) and BabelMulti (3096) targets required some kind of selection, clustering or dimensionality reduction approach. So, given that—at least theoretically—the same information is conveyed by sBNF’s, with a suitably low dimensionality (80), we decided to switch from posteriors to sBNF’s. We were aware that this change may make us pay a high price. Posteriors have a clear meaning, they can be linearly combined and their values suitably fall within the range [0,1], which makes the − log cos(α) distance also range in [0,1] (α being the angle between two vectors of posteriors), with very good results reported in our previous works. On the other hand, bottleneck layer activations have no clear meaning, we don’t really know if they can be linearly combined (e.g. for computing an average query from multiple query instances), and their values are unbounded, so the − log cos(α) distance does no longer apply. Is there any other distance working fine with sBNF? This evaluation poses a great opportunity to address these issues.

2.3. DTW-based search

To perform the search of spoken queries in audio documents, we basically follow the DTW-based approach presented in [3]. In the following, we summarize the approach and the modifications introduced for this evaluation.

Given two sequences of sBNF’s corresponding to a spoken query and an audio document, we first apply VAD to discard non-speech frames, but keeping the timestamp of each frame. To avoid memory issues, audio documents are splitted into chunks of 5 minutes, overlapped 5 seconds, and processed independently. This chunking process is key to the speed and feasibility of the search procedure. Let us consider the VAD-filtered sequences corresponding to a query q = (q[1],q[2],...,q[n]) and an audio document x = (x[1],x[2],...,x[m]) of length m and n, respectively. Since sBNF’s (theoretically) range from −∞ to +∞, we define the distance between any pair of vectors, q[i] and x[j], as follows:

\[ d(q[i], x[j]) = -\log \left( 1 + \frac{q[i] \cdot x[j]}{|q[i]| \cdot |x[j]|} \right) + \log 2 \]  

Note that \( d(v, w) \geq 0 \), with \( d(v, w) = 0 \) if and only if \( v \) and \( w \) are aligned and pointing in the same direction, and \( d(v, w) = +\infty \) if and only if \( v \) and \( w \) are aligned and pointing in opposite direction.

The distance matrix computed according to Eq. 1 is normalized with regard to the audio document \( x \), as follows:

\[ d_{norm}(q[i], x[j]) = \frac{d(q[i], x[j]) - d_{min}(i)}{d_{max}(i) - d_{min}(i)} \] (2)

where:

\[ d_{min}(i) = \min_{j=1,...,n} d(q[i], x[j]) \] (3)

\[ d_{max}(i) = \max_{j=1,...,n} d(q[i], x[j]) \] (4)

In this way, matrix values are in the range [0,1] and a perfect match would produce a quasi-diagonal sequence of zeroes. This can be seen as test normalization since, given a query \( q \), distance matrices take values in the same range (and with the same relative meaning), no matter the acoustic conditions, the speaker, etc. of the audio document \( x \).

Note that the chunking process described above makes the normalization procedure differ from that applied in [3], since \( d_{min}(i) \) and \( d_{max}(i) \) are not computed for the whole audio document but for each chunk independently. On the other hand, considering chunks of 5 minutes might be beneficial, since normalization is performed in a more local fashion, that is, more suited to the speaker(s) and acoustic conditions of each particular chunk.

The best match of a query \( q \) of length \( m \) in an audio document \( x \) of length \( n \) is defined as that minimizing the average distance in a crossing path of the matrix \( d_{norm} \). A crossing path starts at any given frame of \( x \), \( k_1 \in [1, n] \), then traverses a region of \( x \) which is optimally aligned to \( q \) (involving \( L \) vector alignments), and ends at frame \( k_2 \in [k_1, n] \). The average distance in this crossing path is:

\[ d_{avg}(q, x) = \frac{1}{L} \sum_{l=1}^{L} d_{norm}(q[i_l], x[j_l]) \] (5)

where \( i_l \) and \( j_l \) are the indices of the vectors of \( q \) and \( x \) in the alignment \( l \), for \( l = 1, 2, ..., L \). Note that \( i_1 = 1, i_2 = m, j_1 = k_1 \) and \( j_2 = k_2 \). The optimization procedure is \( O(n \cdot m \cdot d) \) in time: \( d \) size of feature vectors and \( O(n \cdot m) \) in space. For details, we refer to [3].

The detection score is computed as \( 1 - d_{avg}(q, x) \), thus ranging from 0 to 1, being 1 only for a perfect match. The starting time and the duration of each detection are obtained by retrieving the time offsets corresponding to frames \( k_1 \) and \( k_2 \) in the VAD-filtered audio document.

This procedure is iteratively applied to find not only the best match but also less likely matches in the same audio document. To that end, a queue of search intervals is defined and initialized with \([1, n]\). Let us consider an interval \([a, b]\), and assume that the best match is found at \([a', b']\), then the intervals \([a, a'-1]\) and \([b'+1, b]\) are added to the queue (for further processing) only if the following conditions are satisfied: (1) the score of the current match is greater than a given threshold \( T \) (in this evaluation, \( T = 0.85\)); (2) the interval is long enough (in this evaluation, half the query length: \( m/2\)); and (3) the number of matches (those already found + those waiting in the queue) is less than a given threshold \( M \) (in this evaluation, \( M = 7\)). An example is shown in Figure 1. Finally, the list of matches for each query is ranked according to the scores and truncated to \( N \) highest scores (in this evaluation, \( N = 1000 \), though it effectively applied only in a few cases).
2.4. Calibration and fusion of system scores

The scores produced by our systems are transformed according to a discriminative calibration/fusion approach commonly applied in speaker and language recognition, that we adapted to STD tasks for MediaEval 2013, in collaboration with Alberto Abad, from L2F, the Spoken Language Systems Laboratory, INESC-ID Lisboa. In the following paragraphs, we just summarize the procedure. For further details, see [5].

First, the so-called $g$-norm (query normalization) is applied, so that zero-mean and unit-variance scores are obtained per query. Then, if $n$ different systems are fused, detections are aligned so that only those supported by $k$ or more systems ($1 \leq k \leq n$) are retained for further processing (in this evaluation, we use $k = 2$). To build the full set of trials (potential detections) we assume a rate of 1 trial per second (which is consistent with the evaluation script). Now, let us consider one of those detections of a query $q$ supported by at least $k$ systems, and a system $A$ that did not provide a score for it. There could be different ways to fill up this hole. We use the minimum score that $A$ has output for query $q$ in other trials. In fact, the minimum score for the query $q$ is hypothesized for all target and non-target trials of query $q$ for which system $A$ has not output a detection score. When a single system is considered ($n = 1$), the majority voting scheme is skipped but $g$-norm and the filling up of missing scores are still applied. In this way, a complete set of scores is prepared, which besides the ground truth (target/non-target labels) for a development set of queries, can be used to discriminatively estimate a linear transformation that will hopefully produce well-calibrated scores.

The calibration/fusion model is estimated on the development set and then applied to both the development and test sets, using the BOSARIS toolkit [10][11]. Under this approach, the Bayes optimal threshold, given the effective prior (in this evaluation, $P_{\text{target}} = C_{\text{miss, target}}/(C_{\text{miss, target}} + C_{\text{fa}}(1 - P_{\text{target}})) = 0.001$), would be applied and—at least theoretically—no further tunings would be necessary. In practice, however, if a system yielded a small amount of detections, we would be using hypothesized scores for most of the trials.

As a result, the calibration/fusion model would be poorly estimated and the Bayes optimal threshold (in this evaluation, 6.9) would not produce good results.

3. STD systems

In this evaluation, we have exploited some publicly available Text-to-Speech (TTS) API’s to perform text-in-audio search as audio-in-audio search. This is just a proof-of-concept aimed at overcoming the Out-Of-Vocabulary (OOV) word issue.

We have applied the Google TTS (gTTS) Python library and command-line interface (CLI) tool [12], which provides two different female (es-ES and es-US) voices, and the Cocoa interface to speech synthesis in MacOS [13], which provides 5 different voices (three male, two female) including both European and American Spanish.

In this way, for each textual term, we synthesize 7 spoken queries: $q_1, q_2, \ldots, q_7$. These spoken queries are downsampling to 8 kHz and applied VAD and sBNF extraction as described in Sections 2.1 and 2.2. The longest query is then taken as reference and optimally aligned to the other queries by means of a standard DTW procedure. Let us consider the sequence of VAD-filtered sBNF vectors for the reference query: $q_i$ of length $m_i$, and the sequence corresponding to another synthesized query: $q_j$ of length $m_j$. The alignment starts at $[1, 1]$ and ends at $[m_i, m_j]$ and involves $L$ alignments, such that each feature vector of $q_i$ is aligned to a sequence of vectors of $q_j$. This is repeated for all the synthesized queries, such that we end up with a set of feature vectors $S_j$ aligned to each feature vector $q_j[j]$, for $i = 1, 2, \ldots, m_i$. Then, each $q_i[j]$ is averaged with the feature vectors in $S_j$ to get a single average query, as follows:

$$q_{avg}[j] = \frac{1}{1 + |S_j|} \left( q_i[j] + \sum_{v \in S_j} v \right) \quad j = 1, 2, \ldots, m_j$$

(6)

Finally, the average query obtained in this way is used to search for occurrences in the audio documents, just in the same way (using the same configuration) as we do in the QbE-STD task.
4. Experimental setup and results
Since BUT/Phonexia sBNF extractors operate on 8 kHz signals, all the query and test audio signals have been downsampled to 8 kHz and stored as 16 bit little-endian signed-integer single-channel WAV files. Audio conversion was performed under MacOS, using the following command:

```
afconvert audio.<ext> -o audio_8k.wav
-d LEI1688000 -c 1 -f WAVE
```

where <ext> represents any audio format extension (such as mp3, aac, wav, etc.).

Two different datasets have been provided for training and development of QbE-STD and STD systems in the Albayzin 2018 Search on Speech Evaluation [14]: MAVIR [15] and RTVE [16]. We have not used the training dataset at all, nor the dev1 set of RTVE. Only the dev set of MAVIR and the dev2 set of RTVE have been used to estimate the calibration/fusion models, and later to search for query occurrences. Search has been also performed on the test sets: COREMAH [17], MAVIR and RTVE, applying the calibration/fusion models and the optimal (MTWV) thresholds obtained on development. In the case of COREMAH (for which no development data was provided), MAVIR calibration/fusion models have been used and heuristic thresholding has been applied, by making yes decisions for the 15% of detections with the highest scores.

One primary (pri) and four contrastive (con1, con2, con3 and con4) systems have been submitted to each combination of task (QbE-STD, STD), condition (development, test) and dataset (COREMAH, MAVIR, RTVE). BabelMulti, Fisher-Mono and FisherTri sBNF’s were used for contrastive systems 2, 3 and 4, respectively. The concatenation of the three sBNF’s (with $80 \times 3 = 240$ dimensions) was used as acoustic representation for contrastive system 1. Finally, the primary system was obtained as the discriminative fusion of the four contrastive systems.

Tables 1 and 2 show ATWV/MTWV performance of the five GTTS-EHU systems on the development sets of MAVIR and RTVE for the QbE-STD and STD tasks, respectively. In all cases, ATWV is obtained for the Bayes optimal threshold (6.9). Along with the MTWV score, the MTWV threshold is shown too (in parentheses). As noted above, the systems eventually submitted for MAVIR and RTVE (in all tasks and conditions) were applied the MTWV threshold.

Table 1: ATWV/MTWV performance on the development sets of MAVIR and RTVE for the QbE-STD systems submitted by GTTS-EHU. The ATWV threshold is set to 6.9. Along with the MTWV score, the MTWV threshold is shown in parentheses.

<table>
<thead>
<tr>
<th>MAVIR</th>
<th>ATWV</th>
<th>MTWV (Thr)</th>
<th>MTWV (Thr)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ATWV</td>
<td>MTWV</td>
</tr>
<tr>
<td>con2</td>
<td>0.1291 (4.06)</td>
<td>0.00000</td>
<td>0.4893 (4.75)</td>
</tr>
<tr>
<td>con3</td>
<td>0.1327 (4.12)</td>
<td>0.00000</td>
<td>0.5722 (5.14)</td>
</tr>
<tr>
<td>con4</td>
<td>0.1590 (4.09)</td>
<td>0.00000</td>
<td>0.5227 (5.09)</td>
</tr>
<tr>
<td>con1</td>
<td>0.1278 (4.14)</td>
<td>0.00000</td>
<td>0.5159 (5.13)</td>
</tr>
<tr>
<td>pri</td>
<td>0.1577 (4.59)</td>
<td>0.00000</td>
<td>0.5352 (5.69)</td>
</tr>
</tbody>
</table>

Table 2: ATWV/MTWV performance on the development sets of MAVIR and RTVE for the STD systems submitted by GTTS-EHU. The ATWV threshold is set to 6.9. Along with the MTWV score, the MTWV threshold is shown in parentheses.

<table>
<thead>
<tr>
<th>MAVIR</th>
<th>ATWV</th>
<th>MTWV (Thr)</th>
<th>MTWV (Thr)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ATWV</td>
<td>MTWV</td>
</tr>
<tr>
<td>con2</td>
<td>0.0396 (5.12)</td>
<td>0.00000</td>
<td>0.0933 (5.30)</td>
</tr>
<tr>
<td>con3</td>
<td>0.0463 (4.82)</td>
<td>0.00000</td>
<td>0.0951 (4.89)</td>
</tr>
<tr>
<td>con4</td>
<td>0.0512 (4.31)</td>
<td>0.00000</td>
<td>0.0916 (4.98)</td>
</tr>
<tr>
<td>con1</td>
<td>0.0398 (5.28)</td>
<td>0.00000</td>
<td>0.0843 (5.21)</td>
</tr>
<tr>
<td>pri</td>
<td>0.0464 (5.43)</td>
<td>0.00000</td>
<td>0.0809 (3.91)</td>
</tr>
</tbody>
</table>

5. Conclusions and future work
The QbE-STD and STD results obtained by our systems on the development sets of MAVIR and RTVE may indicate that the search procedure is detecting few occurrences of the queries, yielding high miss error rates and making it difficult the estimation of good (robust) calibration/fusion models, since most of the trials are missing from system output and we have to hypothesize scores for them. To get a larger amount of target and non-target detections, three of the parameters of the DTW-based search must be relaxed: the maximum amount $M$ of hits per audio chunk (currently, $M = 7$), the minimum score $T$ required to keep searching (currently, $T = 0.85$) and the maximum number $N$ of detections per query for the whole set of audios (currently, $N = 1000$).

QbE-STD detection scores, though badly calibrated, seem to work fine for RTVE but not so well for MAVIR. The difference in performance might be related to a higher acoustic variability or more adverse conditions (reverberation, noise, etc.) for MAVIR.

The trick of synthesizing spoken queries from textual terms to perform STD as QbE-STD seems to be failing, with two possible causes: (1) an acoustic mismatch between the synthesized queries and the test audios might lead to low scores and block the iterative DTW detection procedure; and (2) the use of bottleneck layer activations as frame-level acoustic representation might be incompatible with the query averaging procedure (which worked fine with phone posterior).

Future developments may involve some sort of data augmentation in QbE-STD, such as the use of pseudo-relevance feedback, that is, the use of top matching query occurrences as additional examples. Also, though query averaging is computationally cheap, using it with sBNF representations might be unfeasible and other more expensive strategies to take advantage of the synthesized queries should be explored, such as carrying out multiple searches and fusing the results [18]. Alternatively, we could return to posteriors, by combining the high-dimensional sets of posteriors provided by BUT/Phonexia networks with some sort of feature clustering or feature selection approach.

6. Acknowledgements
We thank Javier Tejedor and Doroteo T. Toledano for organizing a new edition of the Search on Speech Evaluation and for their help during the development process. We also thank Eduardo Lleida and the ViVoLab team for the huge effort of collecting and annotating RTVE broadcasts for the ALBAYZIN 2018 evaluations. This work has been partially funded by the UPV/EHU under grant GIU16/68.
7. References


Cenatav Voice Group System for Albayzin  
2018 Search on Speech Evaluation  
Ana R. Montalvo, Jose M. Ramirez, Alejandro Roble and Jose R. Calvo  
Voice Group, Advanced Technologies Application Center, CENATAV, Havana, Cuba  
{amontalvo, jsanchez, aroble, jcalvo}@cenatav.co.cu  

Abstract  
This paper presents the system employed in the Albayzin 2018 “Search on Speech” Evaluation by the Voice Group of CENATAV. The system used in the Spoken Term Detection (STD) task consists of an Automatic Speech Recognizer (ASR) and a module to detect the terms. The open source Kaldi toolkit is used to build both modules. ASR acoustic models are based on DNN-HMM, S-GMM or GMM-HMM, trained with audio data provided by the organizers and other obtained from ELDA. The lexicon and trigram language model are obtained from the text associated to the audio. The ASR generates the lattices and the word alignments required to detect the terms. Results with development data shown that DNN-HMM model brings up a behavior better or similar to obtained in previous challenges.  

Index Terms: Spoken Term Detection, Automatic Speech Recognition, Kaldi  

1. Introduction  
This is the first participation of the Voice Group of Cenatav in the Albayzin Challenges. We participated in the STD task of the Albayzin 2018 Search on Speech Evaluation, developing a system described in the next section. Spoken Term Detection (STD), is defined by NIST [1] as “searching vast, heterogeneous audio archives for occurrences of spoken terms”. Iberian institutions have been conducted researches recently on this task, as shown in previous Albayzin Challenges [2, 3, 4, 5].  

In STD task, a list of written terms must be detected in different audio files. Although many of the terms may belong to the train vocabulary (INV), the term list is known after processing the audio, which provokes a specific treatment of out-of-vocabulary (OOV) words. The words lexicon and language models of the ASR systems employed to detect OOV words, were obtained from texts where a specific OOV terms list, provided by the evaluators were removed, to force the systems to deal with real OOV words.  

The evaluation of the system is performed with three different databases in native Spanish. The first one is the test partition of the MAVIR [6] corpus, the second one is the SoS test partition of RTVE database [7], while the third is known as COREMAH [8] database.  

The system presented for the task consists of two modules: A Large Vocabulary Continuous Speech Recognition (LVCSR) module and a Spoken Term Detection (STD) module. The acoustic modeling and words decoding were implemented using the Kaldi toolkit [9], while the trigram language models were computed with SRILM [10]. Firstly, the test audio files are processed by a LVCSR based on Deep Neural Network. This produces word lattices which contain the most probable transcriptions of the audio files.  

Section 2 describes the details of the implemented system. Section 3 shows the experimental results obtained using the development data set. Finally, Section 4 concludes the paper and describes the work being under development.  

2. Cenatav Voice Group STD Kaldi System  
The following section describes more detailed the proposed system.  

2.1. Large Vocabulary Continuous Speech Recognition (LVCSR) module  
The first module of the system is a Large Vocabulary Continuous Speech Recognizer implemented following an adaptation of s5 WSJ recipe of Kaldi. The acoustic features used are 13 Mel-Frequency Cepstral Coefficients (MFCC) with cepstral normalization (CMVN) to reduce the effects of the channel.  

The training of the acoustic models begins with a flat initialization of context-independent phone Hidden Markov Models (HMM). Then several re-training and alignment of acoustic models to obtain context-dependent phone HMM, are done following the transformation of the acoustic features:  
- The 13-dimensional MFCC features are spliced across +/- 4 frames to obtain 117 dimensional vectors.  
- Then linear discriminant analysis (LDA) is applied to reduce the dimensionality to 40, using context-dependent HMM states as classes for the acoustic model estimation.  
- Maximum likelihood linear transform (MLLT) is applied to the resulting features, making them more accurately modelled by diagonal-covariance Gaussians.  
- Then, feature-space maximum likelihood linear regression (fMLLR) is applied to normalize inter-speaker variability of the features.  

Obtained phone models consist of three HMM states each in a tied-pdf cross-word tri-phone context This GMM-HMM model was trained with 15000 Gaussians and 2129 senones.  

Then the GMM-HMM model is speaker adapted in a sub-space of Gaussian Mixtures (S-GMM), following [11], using fMLLR features and sharing the same Gaussian Model and same Gaussians quantity per state of the model. This S-GMM contains 9000 Gaussians and 7000 branches.  

The GMM-HMM model is used too, as an alignment for training a DNN-based acoustic model (DNN-HMM) following Dan’s DNN implementation, with 2 hidden layers with 300 nodes each. The number of spliced frames was nine to produce 360 dimensional vectors as input to the first layer. The output layer is a soft-max layer representing the log-posteriors of the context-dependent 2129 states. The total number of parameters is 1703600.  

The LVCSR decoder generates word lattices [12] using any of the mentioned models, these lattices contain the most probable transcriptions of the test utterances where the term
search will be performed. These lattices and the transcriptions obtained are the primary input to the STD module.

The lattices are processed using the lattice indexing technique described in [13], where the lattices of all the test utterances are converted from individual weighted finite state transducers (WFST) to a single generalized factor transducer structure in which the start-time, end-time and lattice posterior probability of each word is stored as a 3-dimensional cost. This factor transducer is an inverted index of all word sequences seen in the lattices.

Thus, given a list of terms to detect, a simple finite state machine is created such that it accepts the terms and composes it with the factor transducer to obtain all its occurrences in the tests utterances, along with the utterance ID, start-time, end-time and lattice posterior probability of each occurrence. All those occurrences are sorted according to their posterior probabilities and a YES/NO decision is assigned to each instance.

2.2. Train and development data
Here is a description of the databases used to train and develop our system for the STD task.

• TC-STAR: We obtained free from ELDA-ELRA a set of audios and transcriptions of Spanish partition of TCSSTAR 2005-2007 [14], corresponding to the Evaluation Package database. It contains 26:40 hours of audio and consists of 17163 expressions with 241412 words. This set was used for training the acoustic models and the language model.

• MAVIR: This corpus was provided by the challenge organizers, and corresponds to talks held by the MAVIR consortium in 2006, 2007 and 2008 [6]. The Spanish training data is contained in “SoS2018_training”, contains 4:20 hours and consists of 5 talks segmented in 2400 expressions with 44423 words. This set was used for training the acoustic models and the language model. The MAVIR development data is contained in “SoS2018_development (1)” and is about one hour in two talks.

• RTVE: The Challenge organizers provided this corpus and its structure is explained in [15]. The corpus is divided in 4 partitions, a “train” one, two development “dev1”, “dev2” and one “test”. The audio files of the “train” partition do not have human-revised transcriptions. Partition “dev1” contains about 53 hours of audios and their corresponding human-revised transcriptions and can be used for either development or training. So, transcriptions (files “trn”) and audio (files “aac”) of twelve RTVE programs (four programs “20H” and eight programs “La Noche en 24H”), of this partition, were manually segmented and labeled by speaker, by us, in expressions less than 40 seconds to obtain a training dataset of 2013 expressions with 139,983 words with a duration of 13:45 hours. This set was used for training the acoustic models and the language model. The development data is in the partition “dev2” and is about 15 hours in twelve RTVE programs.

During the manual segmentation of transcriptions and audio of RTVE programs, we observed that the transcript files did not include many speech expressions that happen in the audio and also some speaker voices overlapping, typical of the spontaneity and level of improvisation during the conversations among the journalists participating in the programs. We eliminate the voices overlapping when segmenting the audio and its corresponding transcription, however only some not transcribed speech expressions, were transcribed by us when we detected them, listening carefully and replaying many times the audio file, during the segmenting process.

All the textual training material of the three databases was revised and corrected, carry to uppercase, substituting the numbers and acronyms for their transcription and finally grouped in a database of 23029 different words and 44:45 hours of duration.

2.3. Vocabulary and Lexicon
The dictionary used by the LVCSR is composed only by words from the transcriptions of the training data. Multilingual G2P transcriber [16] was used to obtain the phonetic transcription of each word. We obtain a general lexicon of 23029 different words.

2.4. Language models
To train the language model used by the LVCSR, we used only the transcriptions of the training data corpus. It consists of 21575 expressions and 23029 different words. This text has been supplied to the SRILM tool to create an Arpa format, trigram language model with 23002 unigrams, 156778 bigrams and 38628 trigrams.

2.5. INV and OOV Terms
This Challenge evaluation defines two sets of terms for STD task: an in-vocabulary (INV) set of terms and an out-of-vocabulary (OOV) set of terms. The OOV set of terms will be composed by out-of-vocabulary words for the LVCSR system, so these OOV terms must be removed from the system dictionary and consequently from the lexicon and the language model.

Table 1: STD scores for MAVIR and RTVE development corpus.

<table>
<thead>
<tr>
<th>Development set</th>
<th>Kaldi Models</th>
<th>MTWV</th>
<th>ATWV</th>
<th>Pfa</th>
<th>Pmiss</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAVIR</td>
<td>tri3b (GMM-HMM)</td>
<td>0.5705</td>
<td>0.5556</td>
<td>0.00006</td>
<td>0.373</td>
</tr>
<tr>
<td>MAVIR</td>
<td>sgmm2_4 (S-GMM)</td>
<td>0.6045</td>
<td>0.5897</td>
<td>0.00005</td>
<td>0.348</td>
</tr>
<tr>
<td>MAVIR</td>
<td>tri4_nnet (DNN_HMM)</td>
<td>0.5974</td>
<td>0.5946</td>
<td>0.00008</td>
<td>0.322</td>
</tr>
<tr>
<td>RTVE</td>
<td>tri3b (GMM-HMM)</td>
<td>0.2943</td>
<td>0.2939</td>
<td>0.00002</td>
<td>0.690</td>
</tr>
<tr>
<td>RTVE</td>
<td>sgmm2_4 (S-GMM)</td>
<td>0.2889</td>
<td>0.2868</td>
<td>0.00001</td>
<td>0.699</td>
</tr>
<tr>
<td>RTVE</td>
<td>tri4_nnet (DNN_HMM)</td>
<td>0.3303</td>
<td>0.3295</td>
<td>0.00002</td>
<td>0.648</td>
</tr>
</tbody>
</table>
3. Experimental results

Table 1 contains the STD scores obtained with the three proposed models (GMM-HMM, S-GMM and DNN-HMM), using the DEV set of MAVIR and RTVE corpus, evaluating with the NIST STDEval-0.7 tool, provided by the Challenge organizers.

This tool provides the probabilities of False Acceptances (Pfa) and Misses (Pmiss) of the STD system, and two metrics that integrates both probabilities [17]:

- Actual Term Weighted Value (ATWV) that integrates Pfa and Pmiss for each term, and averages over all the terms, representing the term weighted value for a threshold set by the system tuned on development data
- Maximum Term Weighted Value (MTWV) that is the maximum TWV achieved by the system for all possible thresholds, not depending on the tuned threshold, representing an upper bound of the system performance.

Comparing with results obtained with the same MAVIR dataset in Albayzin 2016 spoken term detection evaluation [9], our results, for all the proposed models, are similar to obtained by the best system with the DEV set (first line of table 9 of [5]) and the TEST set (first line of table 10 of [5]). Additionally, proposed DNN-HMM model surpasses the behavior of the best system in that evaluation.

However, results shown that the behavior of the evaluated methods is worst with the RTVE set, probably by:

- the differences between the speech of the training set and their transcriptions, explained above, and
- the differences in spontaneity and level of improvisation between MAVIR and RTVE sets.

4. Conclusions and future work

Due to the inexperience of the participants in the use of the Kaldi tool, the lack of the necessary time, and difficulties in the exploitation of the computational resources that we have, it was impossible for us to carry out, before the dead line, the evaluation of the proposed systems applying the Kaldi proxy method, for the DEV and TEST set.

Taking into account the best results of DNN-HMM model in the RTVE set, its lowest Pmiss and lowest performance gap between MTWV and ATWV metrics, indication of well calibrated term detection scores, we decided to send as a primary system the one obtained with the DNN-HMM model and as contrastive systems 1 and 2, those obtained with the models S-GMM and GMM-HMM, respectively.

We consider the participation in this Challenge very useful. The experience acquired by all the participants will serve us for next competitions and more importantly, for the development of our research in the fields of ASR and STD.

We propose to continue and conclude the experiments evaluating the Kaldi proxy method and refining the transcripts of the RTVE database samples, to make them available to the Spanish Thematic Network on Speech Technology (RTTH).

5. Acknowledgments

Finally, we would like to thank the University of Zaragoza and the University of Vigo, in the persons of the professors Eduardo Lleida Solano and Carmen Garcia Mateo, for their support in the elaboration of the textual training data of our system.

References


Acknowledgments

Lleida Solano and Carmen Garcia Mateo, for their support in the elaboration of the textual training data of our system.
MLLP-UPV and RWTH Aachen Spanish ASR Systems for the IberSpeech-RTVE 2018 Speech-to-Text Transcription Challenge

Javier Jorge¹, Adrià Martínez-Villaronga¹, Pavel Golik², Adrià Giménez¹, Joan Albert Silvestre-Cerdà¹, Patrick Doetsch², Vicent Andreu Cisnar³, Hermann Ney², Alfons Juan¹ and Albert Sanchis¹

¹ Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València (Spain)
² Human Language Technology and Pattern Recognition, RWTH Aachen University (Germany)
³ Escola Tècnica Superior d’Enginyeria Informàtica, Universitat Politècnica de València (Spain)

{jjorge, amartinez1, agimenez, jsilvestre, ajuan, josanna}@dsic.upv.es
{golik, doetsch, ney}@cs.rwth-aachen.de
vicismar@inf.upv.es

Abstract

This paper describes the Automatic Speech Recognition systems built by the MLLP research group of Universitat Politècnica de València and the HLTPR research group of RWTH Aachen for the IberSpeech-RTVE 2018 Speech-to-Text Transcription Challenge. We participated in both the closed and the open training conditions.

The best system built for the closed condition was an hybrid BLSTM-HMM ASR system using one-pass decoding with a combination of a RNN LM and show-adapted n-gram LMs. It was trained on a set of reliable speech data extracted from the train and dev1 sets using MLLP’s transLectures-UPV toolkit (TLK) and TensorFlow. This system achieved 20.0% WER on the dev2 set.

For the open condition we used approx. 3800 hours of out-of-domain training data from multiple sources and trained a one-pass hybrid BLSTM-HMM ASR system using open-source tools RASR and RETURNN developed at RWTH Aachen. This system scored 15.6% WER on the dev2 set. The highlights of these systems include robust speech data filtering for acoustic model training and show-specific language modeling.

Index Terms: Automatic Speech Recognition, Acoustic Modeling, Language Modeling, Speech Data Filtering.

1. Introduction

This paper describes the joint collaboration between the Machine Learning and Language Processing (MLLP) research group from the Universitat Politècnica de València (UPV) and the Human Language Technology and Pattern Recognition (HLTPR) research group from the RWTH Aachen University for the participation in the IberSpeech-RTVE 2018 Speech-to-Text transcription challenge, that will be held during the IberSpeech 2018 conference in Barcelona, Spain. Our participation consisted of the submission of three systems: one primary and one contrastive for the closed system training condition, and one primary for the open condition.

The rest of the paper is structured as follows. First, Section 2 describes the RTVE dataset that was provided by the organizers of the challenge. Second, in Section 3 we describe the ASR systems we developed under the closed training conditions. Next, Section 4 details the ASR system that participated in the open training conditions. Finally, Section 5 provides a summary of the work and gives some concluding remarks.

2. RTVE database

The RTVE (Radio Televisión Española) database consists of a collection of 15 different TV shows broadcast by the Spanish public TV station between 2015 and 2018. It comprises 569 hours of audio data, from which 460 hours are provided with subtitles, and 109 hours have been human-revised. In addition, the database includes 3 million sentences extracted from the subtitles of 2017 broadcast news at the RTVE 24H Channel.

The database is provided in five partitions: subs-C24H, the 3M text dataset from the RTVE 24H channel; train, that comprises 463 hours of audio data with non-verbatim subtitles from 16 TV shows; and dev1, dev2, test, consisting of 57, 15 and 40 hours from 5, 2 and 8 different TV shows, respectively. dev1 and dev2 sets were provided with manual corrected transcripts, while the test set was used to gauge the performance of the participating systems. It is important to note that there is a certain overlap between dev1, dev2, test and the train set in terms of TV shows.

In order to have available as many training data as possible during the development stage of the closed-condition system, we decided to split dev1 into two subsets: dev1-train, comprising 43 hours of raw audio to be used for acoustic and language model training; and dev1-dev, 15 hours to be used internally as development data to optimize different components of the system. This split was done at the file level, trying to satisfy that dev1-dev have similar size to dev2, and that both dev1 subsets include files from the same TV shows. We used dev2 as a test set to measure the performance of the system on unseen data.

3. Closed-condition system

3.1. Speech data filtering

Under the closed training conditions, it is extremely important to make the most of the provided training data, specially when it is scarce and/or noisy. This is the case of the RTVE database: on the one hand, the train set comprises only 463 hours of audio, which is not much compared with the amount of data used to train current state-of-the-art systems [1, 2]. On the other hand, training data is not provided with verbatim transcripts but with approximate subtitles. This becomes a major concern when using recurrent neural networks for acoustic modeling, as the accuracy drops significantly when using noisy training data. Therefore, a robust speech data filtering procedure becomes a key point to achieve high ASR performance.

After examining some random samples from train, dev1 and dev2 sets, we first-hand checked that the provided subtitles are far from being verbatim transcripts, but also noted the
presence of 1) subtitle/transcription gaps, 2) subtitle files with no timestamps, 3) audio files considerably larger than their corresponding subtitle files, 4) subtitle files covering timestamps that exceed the length of their corresponding audio file, or 5) human transcription errors in dev1, among others.

For all these reasons, we applied the following speech data filtering pipeline. As subtitle timestamps 1) are not reliable in the train set, 2) are not given in dev1, and 3) are given only at speaker-turn level in dev2, we first force-aligned each audio file to its corresponding subtitle/transcript text. We did this using a preexisting hybrid CD-DNN-HMM ASR [3] system in which the search space was constrained to recognize the exact text (no language model was involved in this procedure), with the only freedom of exploring the different word pronunciations given by the lexicon model and of using an optional silence phoneme at the beginning of each word. In this way we computed the best alignment between the input frame sequence to the sequence of HMM states inferred from the subtitle/transcript text. Then, we applied a heuristic post-filtering based on state-level frame occupation and word-level alignment scores: if either an HMM state is aligned to more frames than the observed average state frame occupation + two times the observed standard deviation, or a word whose average alignment score is lower than a given threshold, then the corresponding word alignment is considered noisy and the word is removed. Next, we completely discarded those files in which more than two thirds of the words were filtered out in the previous step. Finally, we built a clean training corpus by joining words into segments whose boundaries were delimited by large-enough silences and deleted words.

Table 1: Number of raw, aligned raw, aligned speech, and filtered speech hours as a result of applying the speech data filtering pipeline to the whole RTVE database.

<table>
<thead>
<tr>
<th></th>
<th>Raw</th>
<th>Aligned Raw</th>
<th>Speech</th>
<th>Filtered Speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>463</td>
<td>438</td>
<td>252</td>
<td>187</td>
</tr>
<tr>
<td>dev1-train</td>
<td>43</td>
<td>31</td>
<td>24</td>
<td>18</td>
</tr>
<tr>
<td>dev1-dev</td>
<td>14</td>
<td>12</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>dev2</td>
<td>15</td>
<td>12</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Overall</td>
<td>535</td>
<td>493</td>
<td>294</td>
<td>218</td>
</tr>
</tbody>
</table>

Table 1 shows the result of applying this speech data filtering pipeline to the train, dev1 and dev2 sets. The second column shows the raw audio length in hours of each set. The third refers to the amount of raw hours that could be aligned to the corresponding subtitle/transcript text by our alignment system. It must be noted that there were some audio files that the system was not capable to align. This happens when none of the active hypotheses can reach the final HMM state at the last time step, due to an excessive histogram pruning or due to an non-matching transcript. For this reason, 42 hours of audio data could not be aligned. The fourth column gives the total amount of aligned speech data after removing non-speech events that were aligned to the silence phoneme. Surprisingly, the original 438 raw hours from the train set were reduced to 252 hours of speech data, i.e. we detected 186 hours of non-speech events. After some manual analysis of the alignments, we found that a significant portion of these 186 hours is explained by non-subtitled speech, whose corresponding audio frames were in practice aligned to the silence phoneme. Finally, the fifth column shows the number of hours of clean speech data after applying the described heuristic post-filtering procedure and after discarding files that show a high word rejection rate. Starting with the original 535 hours of raw audio, we aligned 294 hours of speech, from which we rejected 76 hours of noisy data, ending up with 218 hours of speech suitable for acoustic training.

<table>
<thead>
<tr>
<th></th>
<th>Sentences</th>
<th>Running words</th>
<th>Vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>340K</td>
<td>4.3M</td>
<td>80K</td>
</tr>
<tr>
<td>subs-C24H</td>
<td>3.1M</td>
<td>57M</td>
<td>160K</td>
</tr>
<tr>
<td>RNN-train</td>
<td>1.8M</td>
<td>35M</td>
<td>176K</td>
</tr>
<tr>
<td>dev1-dev</td>
<td>9.9K</td>
<td>160K</td>
<td>13K</td>
</tr>
<tr>
<td>dev2</td>
<td>7.7K</td>
<td>150K</td>
<td>12K</td>
</tr>
</tbody>
</table>

3.2. Acoustic modeling

The acoustic models (AM) used during the development of the MLLP-RWTH-c1-dev_closed system were trained using filtered speech data from train and dev1-train sets, that is, 205 hours of training speech data. We extracted 16-dim. MFCC features augmented by the full first and second time derivatives, resulting in 48-dim. features.

Our acoustic models were based on the hybrid approach [4, 5]. We first trained a conventional context-dependent Gaussian mixture model hidden Markov model (CD-GMM-HMM) with three left-to-right states. The state-tying schema was estimated following a phonetic decision tree approach [6], resulting in 8.9K tied states. The GMM acoustic model was used to force-align the training data. We then trained a context-dependent feed-forward DNN-HMM using a context window of 11 frames, six hidden layers with ReLU activation functions and 2048 units per layer. We used the transLectures-UPV toolkit (TLK) [7] to train both GMM and DNN acoustic models.

Apart from the feed-forward model, we also trained a BLSTM-HMM model [5]. The DNN was used to refine the alignment between input acoustic features and HMM states. We then trained the BLSTM-HMM model using the open source toolkit TensorFlow [8] and TLK. The BLSTM network consisted of four bidirectional hidden layers with 512 LSTM cells per layer and per direction.

In order to increase the amount of training data, the final submitted system (MLLP-RWTH-p_final_closed) was retrained on a total of about 218 hours from sets train, dev1 and dev2.

3.3. Language modeling

Our language model (LM) for the closed condition consists of a combination of several n-gram models and a recurrent neural network (RNN) model. Also, since TV shows of each audio file are known in advance, we performed an LM adaptation at the n-gram model level.

First, we extracted sentences from all .srt and .trn files. Then we applied a common text processing pipeline to normalize capitalization, remove punctuation marks, expand contractions (i.e. sr. → señor) and transcribe numbers. As already mentioned, we split dev1 into two subsets, dev1-train and dev1-dev, in order to include dev1-train in training. Thus, in this section, we will refer to the combination of train and dev1-train simply as train. For LSTM LM training, we concatenated the train and subs-C24H sets into a single training file and removed redundancy by discarding repeated sentences. Also, sentences were shuffled after each epoch to allow better generalization. To carry out TV-show LM adaptation experiments, we randomly extracted 500 sentences of each TV show from the train set to be used as validation data in the adaptation process. Table 2 provides corpus statistics after normalization.

Second, to define our closed-condition system’s vocabulary, we first computed the vocabulary of both train and subs-C24H sets, and then removed singletons, so that language models can properly model unknown word probabilities. After applying these two steps, the resulting vocabulary had 132K words. The out-of-vocabulary ratios of dev1-dev and dev2 sets were 0.36% and 0.53%, respectively.
Table 3: Perplexities of the different LM components.

<table>
<thead>
<tr>
<th></th>
<th>dev1-dev</th>
<th>dev2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) N-gram train</td>
<td>139.6</td>
<td>183.0</td>
</tr>
<tr>
<td>(b) N-gram subs-C24H</td>
<td>161.2</td>
<td>193.4</td>
</tr>
<tr>
<td>(c) N-gram show-specific</td>
<td>184.0</td>
<td>294.3</td>
</tr>
<tr>
<td>(d) N-gram general (a+b)</td>
<td>107.0</td>
<td>147.5</td>
</tr>
<tr>
<td>(e) N-gram adapt (a+b+c)</td>
<td>99.5</td>
<td>139.1</td>
</tr>
<tr>
<td>(f) RNN</td>
<td>92.3</td>
<td>110.7</td>
</tr>
<tr>
<td>(g) RNN+N-gram general (d+f)</td>
<td>78.2</td>
<td>101.8</td>
</tr>
<tr>
<td>(h) RNN+N-gram adapt (e+f)</td>
<td>68.9</td>
<td>99.2</td>
</tr>
</tbody>
</table>

Third, we trained two standard Kneser-Ney smoothed 4-gram LMs on the train and subs-C24H sets using the SRILM toolkit [9]. Rows (a) and (b) of Table 3 show the perplexities obtained with these models on the dev1-dev and dev2 sets. In addition to these two general n-gram LMs, we trained one n-gram LM for each TV show. Row (c) of Table 3 shows the averaged perplexity of the corresponding TV-show-specific LM for each file.

Next, we trained a RNN LM using the Variance Regularization (VR) criterion [10]. This criterion reduces the computational cost during the test phase. Our models were trained on GPU devices using the CUDNNLM toolkit [11]. The network setup was optimized to minimize perplexity on the dev1-dev set. It consisted of a 1024-unit embedding layer and a hidden LSTM layer of 1024 units. The output layer is a 123K-unit softmax, whose size corresponds to the vocabulary size. The perplexities obtained with this network are depicted in Row (f) of Table 3.

Then, the combination of the LMs was done in two steps. Firstly, we performed a linear interpolation of n-gram models. For the general, non-adapted models, we interpolated the LMs estimated on the train and the subs-C24H sets by minimizing the perplexity on dev1-dev [12]. Row (d) of Table 3 shows the perplexities for this particular LM combination. For each show-specific LM, we performed a three-way interpolation: the individual show-specific LM, the train LM and the subs-C24H LM. In this case, interpolation weights were optimized individually for each TV show so that the perplexity was minimized on the corresponding 500-sentence show-specific validation set, similarly to the approach followed in [13, 14]. Secondly, we combined the interpolated n-gram LMs with the RNN LM.

Next, we analyzed the contribution of different LM combinations during search, leaving fixed the acoustic model to the best BLSTM neural network. Specifically, we carried out recognition experiments using (1) the general n-gram LM, (2) the RNN LM, (3) the interpolation of the RNN LM with the general n-gram LM, and (4) the interpolation of the RNN LM with the adapted, show-specific n-gram LMs. Table 5 shows perplexities and WERs for the dev1-dev and dev2 sets over these four different LM setups.

Table 5: Comparison of different language model combinations using the BLSTM-HMM acoustic model in terms of perplexity, WER % and relative WER % improvement.

<table>
<thead>
<tr>
<th></th>
<th>dev1-dev WER</th>
<th>dev2 WER</th>
<th>∆WER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PPL</td>
<td>WER</td>
<td>PPL</td>
</tr>
<tr>
<td>n-gram general</td>
<td>107</td>
<td>26.5</td>
<td>148</td>
</tr>
<tr>
<td>RNN</td>
<td>92</td>
<td>26.2</td>
<td>111</td>
</tr>
<tr>
<td>RNN + n-gram general</td>
<td>78</td>
<td>25.3</td>
<td>102</td>
</tr>
<tr>
<td>RNN + n-gram adapt</td>
<td>69</td>
<td>24.8</td>
<td>99</td>
</tr>
</tbody>
</table>

The best results were obtained with the combination of RNN and n-gram models, showing a consistent 6% relative improvement in both sets over the baseline general n-gram LM. It is worth noting that in terms of WER, the improvement from using adapted models does not translate to dev2. As dev1-dev and dev2 contain different shows with strongly varying amounts of show-specific text data available for training, not all shows benefit from adaptation equally. Anyway, since the adaption does not degrade the system performance, and given the good improvement seen on dev1-dev, we decided to use the combination of RNN LMs plus adapted n-gram LMs for the final system.

Looking at the system outputs, after carrying out error analysis, we realized that our VAD module [15] was discarding a significant amount of speech regions in the audio files. This significantly affected the WER by increasing the number of deletions. For this reason, we decided to explore other audio segmentation approaches and compare its performance in terms of WER. Concretely, we compared the following approaches: (1) our baseline MLLP-UPV VAD system, based on a speech/non-speech GMM-HMM classifier that ranked second in the Albyazin-2012 audio segmentation challenge [15]; (2) The LIUM Speaker Diarization Tools, a VAD system based on Generalized Likelihood Ratio between speech/non-speech Gaussian models [16]; (3) The well-known CMUseg audio segmentation system using the standard configuration [17]; (4) Apply a fast pre-recognition step to segment the audio file by the recognized silences, using the best CD-FFDNN-HMM acoustic model and a pruned version of the general n-gram LM; and (5) Use the segments generated in (4), and apply VAD the system (1) to classify those segments into speech/non-speech. It is important to note that (3) and (4) are not VAD systems but just audio segmenters, so all detected segments are considered speech.
i.e. all audio is passed through to the ASR. Table 6 shows the WER for each of the five audio segmentation/VAD techniques, including the ratio of discarded audio that is dropped by the VAD prior to decoding.

Table 6: Comparison of different audio segmentation/VAD techniques using the BLSTM-HMM acoustic model and the combination of the RNN LM + adapted n-gram LM. Results in WER % and relative WER % improvement and the ratio of dropped audio.

<table>
<thead>
<tr>
<th>Technique</th>
<th>dev1-dev</th>
<th>dev2</th>
<th>ΔWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMUseg (3)</td>
<td>7.1 23.7</td>
<td>3.9 20.8</td>
<td>7.1</td>
</tr>
<tr>
<td>Pre-Recognition (4)</td>
<td>0 22.9</td>
<td>0 20.6</td>
<td>8.0</td>
</tr>
<tr>
<td>+ MLLP-UPV (5)</td>
<td>3.2 22.3</td>
<td>3.3 20.0</td>
<td>10.7</td>
</tr>
<tr>
<td>LIUM (2)</td>
<td>10.9 24.8</td>
<td>5.9 22.4</td>
<td>-</td>
</tr>
<tr>
<td>MLLP-UPV (1)</td>
<td>10.9 24.8</td>
<td>5.9 22.4</td>
<td>-</td>
</tr>
</tbody>
</table>

As we expected, the baseline VAD system (1) was discarding too much segments, as it was too aggressive compared to other techniques. With either (2) or (3) we obtained a consistent improvement. It was further increased up to 8% by using (4). We decided then to combine this segmentation with our baseline VAD system (1), which led us to achieve an 11% relative WER improvement. In absolute terms, we got a 2.4 WER points gain in dev2, with a final WER of 20.0%. This setup constituted our contrastive closed-condition ASR system. We used a pronunciation lexicon with a vocabulary size of 325k with one or more pronunciation variants. The acoustic model takes 80-dim. MFCC features as input and estimates state posterior probabilities for 5000 tied triphone states. The state tying was obtained by estimating a classification and regression tree (CART) on all available training data. Acoustic modeling was done using a bi-directional LSTM network with four layers and 512 LSTM units in each layer. About 30% of activations are dropped in each layer for regularization purpose [22]. During training we minimized the cross-entropy of a network generated distribution in the softmax output layer at aligned label positions using a Viterbi alignment defined over the 5000 tied triphone states of the CART. We used the Adam learning rate schedule [23] with integrated Nesterov momentum and further reduced the learning rate following a variant of the Newbob scheme. We split input utterances into overlapping chunks of roughly 10 seconds and perform an L2 normalization of the gradients for each chunk. With the normalized gradients the network is updated in a stochastic gradient descent manner where batches containing up to 50 chunks are distributed over eight GPU devices and recombined into a common network after roughly 500 chunks have been processed by all devices.

The language model for the single-pass HMM decoding is a 5-gram count model trained with Kneser-Ney smoothing on a large body of text data collected from multiple publicly available sources. Its perplexity on dev1-dev and dev2 is 173.5 and 173.2 respectively. This open-track system has reached a WER of 18.3% and 15.6% on dev1-dev and dev2 without any speaker or domain adaptation or model tuning.

5. Conclusions

In this paper we have presented the description of the three systems that participated in the IberSpeech-RTVE 2018 Speech-to-Text transcription challenge. Two of them, one primary (MLLP-RWTH-p-final_closed) and one contrastive (MLLP-RWTH-c1-dev_closed), were submitted to the closed training conditions, while the other one (MLLP-RWTH-p-prod_open) participated in the open training track. On the one hand, our best development closed-condition ASR system (MLLP-RWTH-c1-dev_closed), consisting of a BLSTM-HMM acoustic model trained on a reliable set of 205 hours of training speech data, and a combination of both RNN and TV-show adapted n-gram language models, achieved a competitive mark of 20.0% WER on the dev2 set. Our final, primary closed-condition ASR system (MLLP-RWTH-p-final_closed) should offer a similar or even better performance as it followed the same system design setup but trained with all available data, including both development sets. On the other hand, our general-purpose open-condition ASR system (MLLP-RWTH-p-prod_open), without carrying out any speaker, domain nor model adaptation of any kind, scored 15.6% WER on the dev2 set.

6. Acknowledgements

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 761758 (X5gon) and the Spanish government’s TIN2015-68326-R (MINECO/FEDER) research project MORE. This work also financed by grant FPU14/03981 from the Spanish Ministry of Education, Culture and Sport. Finally, we would also like to thank our colleagues at RWTH Aachen for many fruitful discussions: Eugen Beck, Tobias Menne and Albert Zeyer.
7. References


Exploring Open-Source Deep Learning ASR for Speech-to-Text TV program transcription

Juan M. Perero-Codoseró1,2, Javier Antón-Martín1,2, Daniel Tapias Merino1, Eduardo López-Gonzalo2, Luis A. Hernández-Gómez2

1Sigma Technologies S.L.
2GAPS Signal Processing Applications Group, Universidad Politècnica de Madrid

{jmperero, janton, daniel}@sigma-ai.com, {eduardo.lopez, luisalfonso.hernandez}@upm.es

Abstract

Deep Neural Networks (DNN) are fundamental part of current ASR. State-of-the-art systems are hybrid models in which acoustic models (AM) are designed using neural networks. However, there is an increasing interest in developing end-to-end Deep Learning solutions where a neural network is trained to predict character/grapheme or sub-word sequences which can be converted directly to words. Though several promising results have been reported for end-to-end ASR systems, it is still not clear if they are capable to unseat hybrid systems.

In this contribution, we evaluate open-source state-of-the-art hybrid and end-to-end Deep Learning ASR under the IberSpeech-RTVE Speech to Text Transcription Challenge. The hybrid ASR is based on Kaldi and Wav2Letter will be the end-to-end framework. Experiments were carried out using 6 hours of dev1 and dev2 partitions. The lower WER on the reference TV show (LM-20171107) was 22.23% for the hybrid system (lowercase format without punctuation). Major limitation for Wav2Letter has been a high training computational demand (between 6 hours and 1 day/epoch, depending on the training set). This forced us to stop the training process to meet the Challenge deadline. But we believe that with more training time it will provide competitive results with the hybrid system.

Index Terms: TV shows Speech-to-Text transcription, ASR systems, Hybrid DNN-HMM, End-to-end Deep Learning.

1. Introduction

Deep Neural Networks (DNN) have become fundamental part of current ASR systems. State-of-the-art approaches are generally hybrid models in which acoustic models (AM) are designed using neural networks to create HMM class posterior probabilities. These HMM-based neural network acoustic models (DNN-HMM) are combined with conventional pronunciation (PM) and language (LM) models [1].

The main limitations in hybrid ASR systems is a high complexity associated to the boot-strapping process for training the DNN-HMM models, requiring phoneme alignments for frame-wise cross entropy, and a sophisticated beam search decoder [2].

Though several approaches are being proposed to overcome these limitations, such as to train without requiring a phoneme alignment, or to avoid the lexicon [3], there is an increasing interest in working towards end-to-end solutions.

In end-to-end deep learning ASR systems [4] a neural network is trained to predict character/grapheme or sub-word sequences which can be converted directly to words, or even word sequences directly. They present the important advantage of integrating conventional separate acoustic, pronunciation and language models (AM, PM, LM) into a unified neural network modeling framework. Using a simplified training process acoustic, pronunciation and language modeling components are integrated to generate the hypothesized graphemes, sub-words or word sequences. This also greatly simplifies the decoding.

Most end-to-end ASR approaches [5] are typically based on a Connectionist Temporal Classification (CTC) framework [6, 7], a Sequence-to-Sequence attention-based encoder-decoder [8, 9] or a combination of both [10].

Several sequence-to-sequence Neural Network models have been proposed as Recurrent Neural Network Transducer (RNN-T) [11], Listen, Attend and Spell (LAS) [9], and Mono- tonic Alignments [12]. In attention-based encoder-decoder schemes [4] the listener encoder module plays a similar role that a conventional acoustic model, the attender learns alignments between the source and the target sequence, and the decoder works as a language model. Better performance has been reported [4] by modelling longer units such as word pieces models (WPM) and using Multi-head attention (MHA).

Several research works have also been proposed aiming to re-use HMM-based models to improve end-to-end systems. For example, well-trained tied-triphone acoustic models (AM) can be used as an initial model for a character-based end-to-end system [5] or training tied-triphone CTC models from scratch, but in this case a lexicon was required. However, these methods have important limitations as they demand a complex system development, high computation and a large amount of data for training, thus losing the attractiveness of end-to-end systems.

Therefore, even with the promising results already reported for many end-to-end ASR systems, it is still not clear if they are capable to unseat the current state-of-the-art hybrid DNN-HMM ASR systems.

In this paper, our aim is to contribute to the research towards the development of end-to-end ASR systems as an alternative to state-of-the-art hybrid ASR systems. For this purpose, we will develop and compare two open-source hybrid and end-to-end ASR systems for a specific speech-to-text task. As hybrid system the DNN-HMM Kaldi Toolkit [13] will be used, while Wav2Letter [14] will be the end-to-end framework. The speech-to-text task we will work on will be the RTVE IberSpeech 2018 Challenge 1. This task represents a highly demanding domain corresponding to the automatic transcription of TV shows and broadcast news, in specific conditions, as different noisy environments and with the lack of accurate transcriptions for training.

The rest of the paper is structured as follows. In Section

we describe the end-to-end and the hybrid ASR systems, which we have evaluated in the RTVE IberSpeech 2018 Challenge. Section 3 details the datasets we have used: how we have preprocessed them and the experimental protocols we have followed. Results are shown and discussed in Section 4. Finally, we summarize our results and conclusions in Section 5.

2. Deep Learning ASR Systems

2.1. End-to-end Speech Recognition System

As representative of open-source end-to-end ASR systems, we have chosen Wav2Letter. The Wav2Letter ASR system is based on a neural network architecture composed of convolutional units [15], with a Gated Linear Units (GLUs) implementation [16, 17]. The acoustic modeling is based on Mel-Frequency Spectral Coefficients (MFSC) which feed the Gated CNNs that generate letter scores at their outputs. These scores are processed by an alternative to CTC, the Auto Segmentation Criterion (ASG) leading to letter-based sequences (see Figure 1). In order to train the acoustic model, the feature extraction module computes 40-dimensional MFSCs, due to robustness to small time-warping distortions, as referred in [17].

As commented before, the Neural Network architecture is trained to infer the segmentation of each letter in the training transcriptions using Auto Segmentation Criterion (ASG), an alternative criterion to Connectionist Temporal Classification (CTC). CTC takes into account all possible letter sequences, allowing a special blank state, which represents possible garbage frames between letters or the separation between repeated letters. In ASG blank states are replaced by the number of repetition of the previous letter, consequently a simpler graph is obtained [14]. Besides this graph that scores letter sequences depicting the right transcription, another graph is used to score of all letter sequences. Finally, a beam-search decoder (as described in [14]), is used at the last stage. It depends on a beam thresholding, histogram pruning and an optional language model.

2.2. Hybrid Speech Recognition System

In order to compare with end-to-end ASR, we have built a hybrid ASR system using open-source Kaldi Toolkit [13]. The ASR architecture consists of the classical sequence of three main modules: an acoustic model, a dictionary or pronunciation lexicon and a N-gram language model. These modules are combined for training and decoding using Weighted Finite-State Transducers (WFST) [18]. The acoustic modeling is based on Deep Neural Networks and Hidden Markov Models (DNN-HMM).

For the implementation of Kaldi DNN-HMM acoustic modelling we followed the so-called chain model [19], based on a subsampled time-delay neural network (TDNN) [20]. This implementation uses 3-fold reduced frame rate at the output of the network; this represents a significant reduction in decoding computation and the corresponding test time. The reduced frame rate requires HMM traversable in one transition; we use fixed transition probabilities in the HMM, and don’t train them. Additionally, training DNN-HMM following a sequence-level objective function allowed its implementation as a maximum mutual information (MMI) criterion without lattices on GPU: doing a full forward-backward on a decoding graph derived from a phone n-gram language.

Starting from the available transcript of the training speech data, training acoustic models is an iterative process of audio re-alignment starting from GMM/HMM monophone models and progressing to more accurate triphone models through re-training. For all of our experiments we used conventional feature pipe-line that involves splicing the 13-dimensional front-end MFCCs across 9 frames, followed by applying LDA to reduce the dimension to 40 and then further decorrelation using MLLT [21]. For initial GMM/HMM alignments speaker independent acoustic models were obtained using FMLLR. The input features to the neural network in DNN-HMM models was represented by a fixed transform that decorrelates a vector of (40*7)-dimensional features obtained by packing seven frames of 40-dimensional features MFCC(spliced) + LDA+MLLT+FMLLR corresponding to 3 frames on each side of the central frame.

To improve robustness mainly on speakers variability, speaker adaptive training (SAT) based on i-vectors was also implemented [22]. Speaker adaptive models were obtained by fine-tuning DNNs to a speaker-normalized feature space. On each frame a 100-dimensional i-vector is appended to the 40-dimensional acoustic space. In this extended acoustic space i-vectors may supply information about different sources of variability as speakers ID, so the network itself can do any feature normalization that is needed. To overcome some issues reported when test signals have substantially different energy levels than the training data, in our experiments the test-signal energies were energy-normalized to the average of the training data.

3. Experimental Setup

3.1. Datasets

3.1.1. RTVE2018 Database

In this evaluation, we investigate the performance of end-to-end and hybrid ASR systems on RTVE voice contents, a collection of TV shows and broadcast news from 2015 to 2018. Training partition consists of audio files with subtitles, with the following limitations:

• Subtitles have been generated through a re-speaking procedure that sometimes summarizes what has been said, producing imprecise transcriptions.
• Transcriptions have not been supervised by humans.
• Timestamps are not properly aligned with the speech signal.

Trying to avoid the use of these low-quality transcriptions, which could cause confusion in the acoustic space, audio data was initially aligned by a baseline alignment system. This system was the same hybrid system described in Section 2.2. but trained using our own labeled databases, explained in Subsection 3.1.2. To improve the quality of these automatic transcriptions, they were undergone to a manual supervision process.

2 http://catedrartve.unizar.es/rtve2018/RTVE2018DB.pdf
RTVE training partition consists of 460 hours, however due to our limitations in the manual supervision process, only two training datasets have been prepared for our experiments: RTVE_train350 (350 hours of train set) and RTVE_train100 (a RTVE_train350 subset of 100 hours). Validation datasets were extracted from the 10% training set for each RTVE_train350 and RTVE_train100 partitions. These validation datasets have been designed trying to cover the different scenarios in the whole RTVE training data: political and economic news, in-depth interviews, debate, live magazines, weather information, game and quiz shows. Consequently, for testing purposes, two development datasets have been defined as follows:

- RTVE_dev1: 5 hours have been selected in a balanced way in order to have an hour of each show type (e.g. 20H_dev1 is one hour of 20H program).
- RTVE_dev2: 1 hour has been selected carrying out the same procedure as RTVE_dev1.

### 3.1.2. Other Databases

Acoustic models have also been evaluated over open training condition, that is: by using additional datasets. To this end two additional datasets have been used to train the system.

- VESLIM: It consists of 103 hours of Spanish clean voice, where speakers read some sentences. More details in [23].
- OWN MEDIA: It contains 162 hours of TV programs, interviews, lectures and similar multimedia contents. This dataset contains manual transcriptions.

### 3.2. Training

From the hybrid ASR system, our AM was trained following a process based on the Switchboard Kaldi recipe (TDNN Chain models).

In order to create the PM for hybrid ASR system, it is used a set of 29 real phonemes (without silence phones). Instead, end-to-end system, the vocabulary contains 38 graphemes representing the standard Spanish alphabet plus stressed vowels, the apostrophe, symbols for repetitions and separation.

### 3.3. Resources

Experiments have been carried out using several computation resources. A server with 2 Xeon E5-2630v4, 2.2GHz, 10C/20TH and 3 GPUs Nvidia GTX 1080 Ti was used for hybrid ASR system. GPU for the DNN training and CPU for the HMM training and final decoding.

In order to train end-to-end models, more RAM was required, so it was used a GPU Nvidia Quadro P5000 (16 GB), for training and letter decoding.

### 4. Results

#### 4.1. Hybrid ASR System

First, we compared the performance of our hybrid system increasing training data volume from 100 to 350 hours, and by adding additional training data from our external datasets.

Evaluation plans mentioned that a reference TV show (LM-20171107) has been used to obtain results with some commercial systems. This show is a live magazine covering Spanish current events and it has been used to obtain first results. As expected, more than 14% relative improvement in WER is obtained we adding all the available data.

<table>
<thead>
<tr>
<th>Hybrid systems</th>
<th>WER(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTVE_train100 + LM_subtitles</td>
<td>26.01</td>
</tr>
<tr>
<td>RTVE_train350 + LM_subtitles</td>
<td>24.21</td>
</tr>
<tr>
<td>RTVE_train350 + LM_supervised</td>
<td>25.95</td>
</tr>
<tr>
<td>RTVE_train350 + LM_subsuperv</td>
<td>23.47</td>
</tr>
<tr>
<td>RTVE_train350 + Others + LM_subsuperv</td>
<td>23.20</td>
</tr>
<tr>
<td>RTVE_train350 + Others + LM_open</td>
<td>22.23</td>
</tr>
</tbody>
</table>

For testing on closed training condition, we used the full volume of RTVE training data that has been manually revised (RTVE_train350).

In addition to AM, LM has been trained with a different corpus. Four LMs were generated: LM_subtitles (based on subtitles given in RTVE database for the Challenge), LM_supervised (based on transcriptions of RTVE data training supervised by humans), LM_subsuperv (based on two mentioned corpus) and LM_open (based on several corpus: news between 2015 and 2018, interviews, film captions and the two mentioned before).

As shown in Table I, adding supervised transcriptions from the training set to a subtitles-based LM, we achieved a WER of 23.47%, a 3% relative improvement over the same system using a language model trained only with subtitles. This improvement is mainly due to the fact that supervised transcriptions contains some conversational language features (i.e. false starts, truncated words, filler words, syntactic structure changes at talking time, etc.) that are generally omitted in subtitles because of re-speaking procedure. In contrast, a LM only trained with supervised transcriptions did not provide better results. In this case, there were some tags in these transcriptions when words could not be confidently revised/transcribed (e.g. foreign names, mispronunciation, background noise, etc.). Inserting tags meant to include "unk" symbol to the LM and results were not as good as expected.

We next evaluated the systems over open training condition, where we increased the amount of data for both AM and LM training. We combined RTVE training dataset and our own databases (see Section 3.1.2), resulting in a total over 600 hours of speech. As it can be seen in Table I the hybrid system provided a slight improvement. But, more importantly, WER went down to 22.23% when transcriptions from additional corpus were incorporated to train a LM. As a result, increasing both the amount of audio and transcription data will enable us to obtain the maximum performance, and to cover as much information as possible appearing in test files.

The division of development datasets according to the different show types makes it possible a deeper error analysis. Table II shows that models applied to TV programs as 20H and Millennium obtain the best results, a low WER of 14-17% for the best models. This could be explained because contents are daily news having good acoustic conditions (clean voice, only one speaker at time) and being better featured in LM. However, CA (Comando Actualidad) dataset contains some challenging scenarios (interviews at the street, background noise, overlapping, music). As a result, models achieve a high value of WER (49.51%).

Furthermore, it has to be emphasized that reference master of transcriptions was given without any review from our part. To evaluate the possible impact of transcription errors in the refer-

Table I: WER on reference TV (LM-20171107) show for acoustic models over closed and open training conditions (different train data sizes and language models).
Table II: WER(%) on the different datasets of models over a closed and open training conditions (different train data volume and language models). The duration of each dataset is one hour.

<table>
<thead>
<tr>
<th>Hybrid systems</th>
<th>20H_dev1</th>
<th>AP_dev1</th>
<th>CA_dev1</th>
<th>LM_dev1</th>
<th>Mill_dev1</th>
<th>LN24H_dev1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTVE_train100 + LM_subtitles</td>
<td>17.67</td>
<td>24.36</td>
<td>51.95</td>
<td>25.41</td>
<td>19.59</td>
<td>27.47</td>
</tr>
<tr>
<td>RTVE_train350 + LM_subtitles</td>
<td>16.04</td>
<td>21.08</td>
<td>51.58</td>
<td>23.61</td>
<td>17.43</td>
<td>26.35</td>
</tr>
<tr>
<td>RTVE_train350 + LM_supervised</td>
<td>16.05</td>
<td>22.18</td>
<td>59.34</td>
<td>25.06</td>
<td>19.22</td>
<td>26.75</td>
</tr>
<tr>
<td>RTVE_train350 + LM_subsuperv</td>
<td>15.38</td>
<td>21.43</td>
<td>49.67</td>
<td>22.86</td>
<td>17.81</td>
<td>25.15</td>
</tr>
<tr>
<td>RTVE_train350 + Others + LM_subsuperv</td>
<td>15.21</td>
<td>21.95</td>
<td>49.51</td>
<td>22.30</td>
<td>18.51</td>
<td>25.09</td>
</tr>
<tr>
<td>RTVE_train350 + Others + LM_open</td>
<td><strong>14.88</strong></td>
<td><strong>20.94</strong></td>
<td>49.55</td>
<td><strong>21.44</strong></td>
<td><strong>17.01</strong></td>
<td><strong>24.13</strong></td>
</tr>
<tr>
<td><strong>End-to-end system</strong></td>
<td><strong>RTVE_train100</strong></td>
<td>71.61</td>
<td>75.38</td>
<td>87.06</td>
<td>72.35</td>
<td>69.49</td>
</tr>
</tbody>
</table>

Figure 2: Comparison in loss function terms during the end-to-end models training, on 100 hours of various databases: LibriSpeech, VESLIM and RTVE_train100.

5. Conclusions

In this contribution we have tested the use of open-source hybrid and end-to-end ASR systems under the RTVE IberSpeech 2018 Challenge. According to our results, the development of a hybrid DNN-HMM Kaldi Toolkit [13] seems to be capable to address the difficulties of this hard task but it requires to put more effort in obtaining better quality transcriptions to improve both acoustic and language models. It is important to remark that WER of 8.51% is obtained, in the best conditions. However, further research on the use of robust feature spaces and DNN training should be addressed looking for robustness in the more challenging scenarios (street interviews with background noise, speakers overlapping, music, etc.). In what relates to end-to-end Wav2Letter system, we must acknowledge that our research has been limited by a high training computational time. Nevertheless, even under this limitation, we found that when compared to read speech (as LibriSpeech) it seems that both, the lack of correct transcriptions and the difficulties of dealing with conversational and noisy speech, will require more research so this kind of end-to-end architectures could be used for these challenging tasks.

6. References


The Vicomtech-PRHLT Speech Transcription Systems for the IberSPEECH-RTVE 2018 Speech to Text Transcription Challenge

Haritz Arzelus1, Aitor Álvarez1, Conrad Bernath1, Eneritz García1, Emilio Granell2, Carlos-D. Martínez-Hinarejos2

1Vicomtech, Human Speech and Language Technology Group, Spain
2Pattern Recognition and Human Language Technologies Research Center, Universitat Politècnica de València, Spain

[harzelus,aalvarez,cbernath,egarciam]@vicomtech.org, [egranell,cmartine]@dsic.upv.es

Abstract

This paper describes our joint submission to the IberSPEECH-RTVE Speech to Text Transcription Challenge 2018, which calls for automatic speech transcription systems to be evaluated in realistic TV shows. With the aim of building and evaluating systems, RTVE licensed around 569 hours of different TV programs, which were processed, re-aligned and revised in order to discard segments with imperfect transcriptions. This task reduced the corpus to 136 hours that we considered as nearly perfectly aligned audios and that we employed as in-domain data to train acoustic models.

A total of 6 systems were built and presented to the evaluation challenge, three systems per condition. These recognition engines are different versions, evolution and configurations of two main architectures. The first architecture includes an hybrid LSTM-HMM acoustic model, where bidirectional LSTMs were trained to provide posterior probabilities for the HMM states. The language model corresponds to modified Kneser-Ney smoothed 3-gram and 9-gram models used for decoding and re-scoring of the lattices respectively. The second architecture includes an End-To-End based recognition system, which combines 2D convolutional neural networks as spectral feature extractor from spectrograms with bidirectional Gated Recurrent Units as RNN acoustic models. A modified Kneser-Ney smoothed 5-gram model was also integrated to re-score the E2E hypothesis. All the systems’ outputs were then punctuated using bidirectional RNN models with attention mechanism and capitalized through rescasing techniques.

Index Terms: speech recognition, deep learning, end-to-end speech recognition, recurrent neural networks

1. Introduction

The IberSPEECH-RTVE 2018 Speech to Text Transcription Challenge calls for Automatic Speech Recognition (ASR) systems that are robust against realistic TV shows. It is a notable trend that aims to approach ASR technology to different applications such as automatic subtitling or metadata generation in the broadcast domain. Although most of this work is still performed manually or through semiautomatic methods (e.g. re-speaking), the current state of the art in speech recognition suggests that this technology could start to be exploitable autonomously without any post-edition task, mainly on contents with optimal audio quality and clean speech conditions.

The use of Deep Learning algorithms in speech processing have made it possible to introduce this technology in such a complex scenario through the use of systems based on Deep Neural Networks (DNNs) or more recent architectures based on the End-To-End (E2E) principle.

Historically, ASR systems have made use of Hidden Markov Models (HMMs) to capture the time variability of the signal and Gaussian Mixture Models (GMMs) to model the HMM state probability distributions. However, numerous works have shown that DNNs in combination with HMMs can outperform traditional GMM-HMM systems at acoustic modeling on a variety of datasets [1]. More recently, new attempts have been focused on building E2E ASR architectures [2], which directly map the input speech signal to grapheme/character sequences and jointly train the acoustic, pronunciation and language models as a whole unit [3, 4, 5, 6]. Nowadays, two main approaches predominate to train E2E ASR models. On the one hand, the Connectionist Temporal Classification (CTC) is probably the most widely used criterion for systems based on characters [2, 7, 8], sub-words [9] or words [10]. It employs Markov assumptions and dynamic programming to efficiently solve sequential problems [2, 3, 7]. On the other hand, attention-based methods employ an attention mechanism to perform alignment between acoustic frames and characters [4, 5, 6]. Unlike CTC, it does not require several conditional independence assumptions to obtain the label sequence probabilities, allowing extremely non-sequential alignments. Additionally, a number of enhancement techniques have been employed to overcome the performance of these systems, such as Data Augmentation [11], Transfer Learning [12], Dropout [13] or Curriculum Learning [14], among others.

Our systems were constructed following both DNN-HMM and E2E architectures basis, given that depending on the available training data, one approach performed more robustly than the other on the development set. A total of 6 systems were presented to the evaluation challenge, three systems per condition (closed and open). Regarding the closed condition, in which only the data released by RTVE could be used to train and evaluate systems, two systems based on the DNN-HMM and one E2E system were presented. In contrast, the results obtained from two E2E based systems and one DNN-HMM system were submitted for the open condition. In all systems, the raw recognized output was punctuated, capitalized and normalized automatically using Recurrent Neural Models (RNNs), rescasing techniques and rule-based heuristics respectively.

2. Corpus processing

Depending on the training condition, different datasets were used to train and evaluate the ASR systems.
2.1. RTVE2018 dataset

The RTVE2018 dataset was released by RTVE and comprises a collection of TV shows drawn from diverse genres and broadcast by the public Spanish National Television (RTVE) from 2015 to 2018. The real number of hours provided in the original dataset as training and development sets is presented in Table 1.

Table 1: Duration of each partition of the original RTVE2018 dataset

<table>
<thead>
<tr>
<th>subset</th>
<th>duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>462 h. 9 min.</td>
</tr>
<tr>
<td>dev1</td>
<td>62 h. 23 min.</td>
</tr>
<tr>
<td>dev2</td>
<td>15 h. 13 min.</td>
</tr>
<tr>
<td>total</td>
<td>539 h. 45 min.</td>
</tr>
</tbody>
</table>

The main problem of this dataset was that a great amount of audios had imperfect transcriptions and, therefore, they could not be used as such for training and evaluation purposes. With the aim of recovering only the correctly aligned segments, a highly costly process was carried out, where an alignment and re-alignment techniques were performed first and a manual revision and automatic recognition task afterwards. The alignment and re-alignment processes consist of two steps. In the first step, we tried to align the original audios with their corresponding transcriptions using 4 different beam values (10; 100; 1,000 and 10,000). For the case of the train partition, a total of 101 hours and 47 minutes were only aligned after this initial step. Thus, 360 hours and 22 minutes were definitely discarded to be used in any training process. A second step of re-alignment was then performed over this new subset of 101 hours and 47 minutes, using beam and retry-beam values of 1 and 2 respectively, obtaining a total of 86 hours and 29 minutes of audio segments that were considered as nearly correctly aligned. The alignments processes were performed using a feed-forward DNN-HMM acoustic model trained with the Kaldi toolkit [15] and estimated over contents from the broadcast domain. Finally, a small partition of the nearly correctly aligned hours were revised manually, whilst the remaining were recognized using a different recognition architecture as employed for the alignments. In this case, the recognition was performed using an E2E based recognition system trained with the same contents from the broadcast domain. Only the recognition outputs that fit exactly to the reference were tagged as perfect segments. The same cleaning methodology was also applied on the dev1 and dev2 partitions.

The total number of hours discarded after the first step, and the hours tagged as nearly perfect and completely perfect are summarized in Table 2. These hours correspond to audio segments that lasted more than one second, since the shorter ones were also discarded.

Table 2: RTVE2018 dataset after the alignment and re-alignment processes

<table>
<thead>
<tr>
<th>subset</th>
<th>alignment (discarded)</th>
<th>re-alignment (nearly perfect)</th>
<th>revision+E2E (perfect)</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>360 h. 22 min.</td>
<td>86 h. 29 min.</td>
<td>56 h. 27 min.</td>
</tr>
<tr>
<td>dev1</td>
<td>7 h. 43 min.</td>
<td>44 h. 34 min.</td>
<td>29 h. 39 min.</td>
</tr>
<tr>
<td>dev2</td>
<td>9 h. 27 min.</td>
<td>5 h. 8 min.</td>
<td>4 h. 3 min.</td>
</tr>
<tr>
<td>total</td>
<td>377 h. 32 min.</td>
<td>136 h. 11 min.</td>
<td>90 h. 9 min.</td>
</tr>
</tbody>
</table>

As it can be seen in Table 2, a high number of hours were discarded from the original RTVE2018 dataset. In the end, a total of 136 hours and 11 minutes were considered as nearly perfect audios, whilst only 90 hours and 9 minutes can be thought to be perfectly aligned including the train, dev1 and dev2 re-aligned partitions. These both subsets were finally used to build and tune the acoustic models (AM). The development set for the tuning of the systems was extracted from the completely perfect subset, as it is shown in Table 3.

Table 3: New perfectly and nearly perfectly aligned subsets for train and development. The 4 hours from dev correspond to the same contents in both subsets.

<table>
<thead>
<tr>
<th>subset</th>
<th>train</th>
<th>dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfectly aligned</td>
<td>86 h. 9 min.</td>
<td>4 h.</td>
</tr>
<tr>
<td>Nearly perfect aligned</td>
<td>132 h. 11 min.</td>
<td>4 h.</td>
</tr>
</tbody>
</table>

In terms of text data, a total of 3.5 million sentences and 61 million words were compiled. This data was used to estimate the language models (LM) and the punctuation and capitalization modules.

2.2. Open dataset

The open dataset was used to build the ASR systems for the open condition. In addition to the perfectly re-aligned subset from the RTVE2018 dataset, 4 different corpora were prepared for training. The SAVAS corpus [16] is composed of broadcast news contents from the Basque Country’s public broadcast corporation EiTB (Euskal Irri Telebista), and includes annotated and transcribed audios in both clear (studio) and noisy (outside) conditions. The Youtube RTVE Series corpus includes Spanish broadcast contents of RTVE shows and series gathered from the Youtube platform. The audio contents were downloaded along with the automatic transcriptions provided by the platform. These audios and their corresponding automatic transcriptions were then split and re-aligned following the same methodology as it was explained in Section 2.1. Finally, the Albayzin [17] and Multext [18] corpora were also included. The development set corresponded to the in-domain new dev partition shown in Table 3. The total amount of hours available for the open condition are summarized in Table 4.

Table 4: The open dataset description

<table>
<thead>
<tr>
<th>corpus</th>
<th>#hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTVE2018</td>
<td>132 h. 11 min.</td>
</tr>
<tr>
<td>SAVAS</td>
<td>190 h. 38 min.</td>
</tr>
<tr>
<td>Youtube RTVE</td>
<td>197 h. 13 min.</td>
</tr>
<tr>
<td>Albayzin</td>
<td>6 h. 5 min.</td>
</tr>
<tr>
<td>Multext</td>
<td>53 min.</td>
</tr>
<tr>
<td>Total</td>
<td>497 h. 20 min.</td>
</tr>
</tbody>
</table>

Regarding text data, data selection techniques were applied on general news data gathered from digital newspapers and using the LM created with the in-domain RTVE2018 text data as a reference. A total of 3.5 million sentences and 71 million words were selected with a maximum perplexity threshold value of 120. Hence, summing the in-domain and new texts data, a total of 132 million words were employed to estimate the LM, and punctuation and capitalization modules for the open condition.
3. Main architectures

Two main architectures were employed to build the systems for both closed and open conditions.

3.1. LSTM-HMM based systems

These systems include a bidirectional LSTM-HMM acoustic model and n-gram language models for decoding and rescoring purposes. The AMs and final graphs were estimated using the Kaldi toolkit. The AM corresponded to a hybrid LSTM-HMM implementation, where bidirectional LSTMs were trained to provide posterior probability estimates for the HMM states. This model was constructed with a sequence of 3 LSTM layers, using 640 memory units in the cell and 1024 fully connected hidden layer outputs. The number of steps used in the estimation of the LSTM state before prediction of the first label was fixed to 40 in both contexts. Furthermore, modified Kneser-Ney smoothed 3-gram and 9-gram models were used for decoding and re-scoring of the lattices respectively. Both LMs were estimated using the KenLM toolkit.

3.2. E2E based systems

The E2E systems were developed following the Deep Speech 2 architecture [2]. The core of the system is basically an RNN model, in which speech spectrograms are ingested and text transcriptions are provided as output.

Initially, a sequence of 2 layers of 2D convolutional neural networks (CNN) are employed as spectral feature extractors from spectrograms. A 2D batch normalization function is then applied to the output of both layers, in addition to a hard tanh function as an activation function. The E2E systems were set up using 5 layers of bidirectional Gated Recurrent Units (GRU) [20] layers as RNN networks. Each hidden layer is composed of 800 hidden units. After the bidirectional recurrent layers, a fully connected layer is applied as the last layer of the whole model. The output corresponds to a softmax function which computes a probability distribution over the characters. During the training process, the CTC loss function is computed to measure the error of the predictions, whilst the gradient is estimated using backpropagation through time algorithm with the aim of updating the network parameters. The optimizer is the Stochastic Gradient Descent (SGD).

In addition, an external LM was integrated for decoding with the aim of rescoring the initial lattices. To this end, modified Kneser-Ney smoothed 5-grams models were estimated using the KenLM toolkit.

4. Systems descriptions

A total of 6 systems based on the above described architectures were submitted to the challenge, three systems per condition.

4.1. Closed condition

4.1.1. Primary system

The primary system submitted to the closed condition was called ‘Vicomtech-PRHLT_p-K1_closed’ and it is a bidirectional LSTM-HMM based system combined with a 3-gram LM for decoding and a 9-gram LM for re-scoring lattices. The AM was trained for 10 epochs, with an initial and final learning rate of 0.0006 and 0.00006 respectively, using a mini-batch size of 100 and 20,000 samples per iteration. The AM was trained with the nearly perfectly aligned partition (see Table 3), which was 3-fold augmented through the speed based augmentation technique. Each audio was transformed randomly depending on a modification parameter ranged between 0.9 and 1.1 values. A total of 396 hours and 33 minutes were therefore used for training. The LMs were estimated with the in-domain texts compiled from the RTVE2018 dataset.

4.1.2. Contrastive systems

The first contrastive system was called ‘Vicomtech-PRHLT_p-E1_closed’ and it was set up using the same configuration of the primary system, but the AM was estimated using the 3-fold augmented acoustic data of the perfectly aligned partition (see Table 3). A total of 258 hours and 27 minutes were employed for training.

The same data was used to build the the second contrastive system, tagged as ‘Vicomtech-PRHLT_p-E2-E1_closed’. It was an E2E recognition system which follows the architecture described above, and it was evolved for 30 epochs. The LM was a 5-gram with an alpha value of 1.5 and a beam-width of 1000 during decoding.

4.2. Open condition

4.2.1. Primary system

The primary system of the open condition was called ‘Vicomtech-PRHLT_p-E1_open’ and it was based on the E2E architecture described in Section 3.2. This system was an evolution of an already existing E2E model, which was built using the 3-fold augmented SAVAS, Albayzin, and Multtext corpora for 28 epochs. This model reached a WER of 7.2% on a 4 hours test set of the SAVAS corpus.

For this challenge, it was evolved for 2 new epochs using the same corpora in addition to the 3-fold augmented nearly perfectly aligned corpus obtained from the RTVE2018 dataset (see Table 3). A total of 897 hours were used for training. The LM was a 5-gram trained with the text data from the open dataset, with an alpha value of 0.8 and a beam-width of 1000 during decoding.

4.2.2. Contrastive systems

The first contrastive system was called ‘Vicomtech-PRHLT_p-E2-E2_open’ and as the primary system, it was based on the previously explained E2E architecture. This system was also an evolution of the already existing E2E model, but in this case, it was evolved for one epoch using the 3-fold augmented SAVAS, Albayzin, nearly perfectly aligned partition and Youtube RTVE corpora. The duration of the total amount of training audios was 1488 hours. The LM was a 5-gram trained with the text data from the open dataset, with an alpha value of 0.8 and a beam-width of 1000 during decoding.

The second contrastive system was composed by a bidirectional LSTM-HMM acoustic model combined with a 3-gram LM for decoding and a 9-gram LM for re-scoring lattices. The AM was evolved for 10 epochs, with an initial and final learning rate of 0.0006 and 0.00006 respectively, using a mini-batch size of 100 and 20,000 samples per iteration, and it was trained with the same data as the primary system of the open condition. The LMs were estimated with the text data from the open dataset.
5. Results

The results obtained over the development set shown in Table 3 are presented in the following Table 5. The development set is composed by audio segments from all the TV shows included in the original RTVE2018 dataset and lasts a total of 4 hours.

Table 5: WER results for each submitted system over the generated development set

<table>
<thead>
<tr>
<th>type</th>
<th>system</th>
<th>cond.</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Vicomtech-PRHLT_p-K1_closed</td>
<td>Closed</td>
<td>22.6</td>
</tr>
<tr>
<td>C1</td>
<td>Vicomtech-PRHLT_c1-K2_closed</td>
<td></td>
<td>22.8</td>
</tr>
<tr>
<td>C2</td>
<td>Vicomtech-PRHLT_c2-E1_closed</td>
<td></td>
<td>26.6</td>
</tr>
<tr>
<td>F</td>
<td>Vicomtech-PRHLT_p-E1_open</td>
<td>Open</td>
<td>20.7</td>
</tr>
<tr>
<td>C1</td>
<td>Vicomtech-PRHLT_c1-E2_open</td>
<td></td>
<td>20.5</td>
</tr>
<tr>
<td>C2</td>
<td>Vicomtech-PRHLT_c2-K1_open</td>
<td></td>
<td>22.0</td>
</tr>
</tbody>
</table>

5.1. Processing time and resources

The decodings of the 6 recognition systems were performed on an Intel Xeon CPU E5-2683v4 2.10 GHz 4xGPU server with 256GB DDR4 2400MHz RAM memory. Each GPU corresponds to an NVIDIA GeForce GTX 1080 Ti 11GB graphics acceleration card.

The following Table 6 presents the processing time and computational resources needed by each submitted system for the decoding of the released test set of almost 40 hours of audio. It should be noted that the LSTM-HMM based systems were decoded using CPU cores, whilst the E2E systems took advantage of the GPU cards.

Table 6: Processing time and computational resources needed by each submitted system

<table>
<thead>
<tr>
<th>system</th>
<th>RAM</th>
<th>CPU cores</th>
<th>GPU</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vicom-PRHLT_p-K1_closed</td>
<td>12GB</td>
<td>20</td>
<td>-</td>
<td>24h</td>
</tr>
<tr>
<td>Vicom-PRHLT_c1-K2_closed</td>
<td>12GB</td>
<td>20</td>
<td>-</td>
<td>24h</td>
</tr>
<tr>
<td>Vicom-PRHLT_c2-E1_closed</td>
<td>4GB</td>
<td>8</td>
<td>7GB</td>
<td>12h</td>
</tr>
<tr>
<td>Vicom-PRHLT_p-E1_open</td>
<td>5GB</td>
<td>8</td>
<td>7GB</td>
<td>8h</td>
</tr>
<tr>
<td>Vicom-PRHLT_c1-E2_open</td>
<td>5GB</td>
<td>8</td>
<td>7GB</td>
<td>9h</td>
</tr>
<tr>
<td>Vicom-PRHLT_c2-K1_open</td>
<td>19GB</td>
<td>12</td>
<td>-</td>
<td>40h</td>
</tr>
</tbody>
</table>

6. Conclusions

In this paper, the ASR systems submitted to the IberSPEECH-RTVE Speech to Text Transcription Challenge 2018 have been presented. In the beginning, one of the most costly task was the processing of the released RTVE2018 dataset, since a high number of transcriptions were imperfect or do not fit exactly to the related spoken audio. Furthermore, the type of contents posed a notable difficulty to the task, given that the TV shows included most of the main challenges for any speech recognition engine, including spontaneous speech, accents, noise backgrounds, and/or overlapped speakers, among others. Thus, the cleaning process of the dataset became a crucial task to exploit the data correctly.

Looking at the results obtained on the internally generated development test and presented in Table 5, it can be clearly deduced that LSTM-HMM based systems performed better when fewer training data were available. In fact, the primary system in the closed condition achieved an error of 4 percentage points lower than the E2E based second contrastive system. In this condition, it is also remarkable how the primary system, trained with nearly correctly aligned audios, achieved better results than the first contrastive LSTM-HMM based system, which was built with perfectly aligned contents, even if the primary system included more training data. It suggests that in this case, exploiting more data although they were not aligned exactly, helped systems to perform better.

In the open condition, the E2E based systems achieved better results than the LSTM-HMM based one. It could be expected since more training data were available to train models. Even if the first contrastive system obtained a slightly better performance than the primary one, a qualitative evaluation of the results gave as the intuition that the primary system was more robust against spontaneous speech. In this sense, the alpha value (0.8), which defines the weight of the LM against the AM, of the E2E systems were lower than the alpha value (1.5) employed in the E2E system of the closed condition, given that the AM performed better and the global system obtained higher precision, especially with spontaneous speech.

Finally, it should be remarked that all the error rates achieved in this work are lower or at least are in the range of the reference WER values given in the evaluation plan. These WER values were obtained by commercial ASR systems over one TV show in the dataset, and ranged between 22% and 27% of word error rate.

7. References


The Intelligent Voice ASR system for the Iberspeech 2018 Speech to Text Transcription Challenge

Nazim Dugan1, Cornelius Glackin1, Gérard Chollet1, Nigel Cannings1
1Intelligent Voice Ltd, London, UK

nazim.dugan@intelligentvoice.com

Abstract
This paper describes the system developed by the Empathic team for the open set condition of the Iberspeech 2018 Speech to Text Transcription Challenge. A DNN-HMM hybrid acoustic model is developed, with MFCC’s and iVectors as input features, using the Kaldi framework. The provided ground truth transcriptions for training and development are cleaned up using customized clean-up scripts and then realigned using a two-step alignment procedure which uses word lattice results coming from a previous ASR system. 261 hours of data is selected from train and dev1 subsections of the provided data, by applying a selection criterion on the utterance level scoring results. The selected data is merged with the 91 hours of training data used to train the previous ASR system with a factor 3 times data augmentation by reverberation using a noise corpus on the total training data, resulting a total of 1057 hours of final training data. Selected text from the train and dev1 subsections are also used for new pronunciation additions and language model (LM) adaptation of the LM from the previous ASR System. The resulting model is tested using data from the dev2 subsection selected with the same procedure as the training data.

Index Terms: speech recognition, forced alignment, neural network

1. Introduction

Deep Neural Networks (DNNs) and Hidden Markov Model (HMM) based hybrid Automatic Speech Recognition (ASR) systems [1] are still widely used despite the recent rise of end-to-end speech recognition systems [2], based on recurrent neural networks (RNNs). An important part of the ASR acoustic model training procedure is the time alignment of the training audio with the transcriptions. Usually, the utterance start and end times of the training transcriptions are given to the ASR training system, and in the utterance, character or phone level alignments of the training text are computed by an iterative process, or with Connectionist Temporal Classification (CTC) [3]. If the given start and/or end time is wrong for an utterance it would be discarded by the ASR training system or if not, it will reduce the accuracy of the ASR training. Therefore, input data preparation with accurate utterance level time alignments is an important prerequisite for training an ASR system in either case.

Kaldi is a commonly used speech recognition framework which supports DNN-HMM hybrid systems with a couple of different DNN implementations. It also supports Gaussian Mixture Model (GMM) HMM hybrid systems [4] which are mostly used iteratively for the alignment of input transcripts with the input audio frames. State of the art Kaldi recipes use a phonetic description of the transcripts and the GMM-HMM iterative phone alignment methodology, even though there is experimental support for character-based training and CTC alignment in the Kaldi framework. These new experimental recipes mostly produce ASR systems with slightly lower accuracy compared to the state of the art recipes.

In this work, a DNN-HMM hybrid Kaldi recipe with GMM-HMM iterative phone alignment was used for training a European Spanish ASR system with 8kHz down-sampled input audio and transcriptions from the Iberspeech 2018 speech to text transcription challenge training and development data, and four well known European Spanish corpora. A previous ASR system for European Spanish language was used for the utterance level time alignments of the input transcriptions provided for the Iberspeech 2018 speech to text transcription challenge, training and development. These transcriptions are also used for the language model (LM) adaptation of the previous ASR model using the SRILM toolkit [6].

2. Data preparation

It was observed that the provided utterance level time alignments of the training and development transcripts were not accurate and some of the provided files do not have any time alignments. Word lattice results coming from a previous ASR system trained using four well known European Spanish corpora: Albayzin [7], Dihana [8], CORLEC-EHU [9], TCSTAR [10] and the text corpus El País were used for the re-alignment of the training and development transcripts. The ASR system used to generate the word lattice was developed with the same input data as the ASR model described in [5]. However, an improved recipe of the Kaldi framework [11] is used which utilizes a factor 3-times data augmentation with reverberation and noise addition, high resolution Mel Frequency Cepstral Coefficients (MFCCs), iVectors [12] and nnet3 chain implementation. This previous ASR model will be referred to as the base model throughout the remaining text.

The provided training and development audio files are sub-sampled to 8kHz before being used in the training and testing processes.
2.1. Two-step time alignment procedure

A two-step utterance level time alignment procedure is used which includes forced alignment of plain text transcripts and word level time alignments using the NIST sclite ASR scoring utility in Speech Recognition Scoring Toolkit (SCTK) [13].

2.1.1. Forced alignment of plain text transcripts

Time alignments in the sr type subtitle and stm type reference files of the train, dev1 and dev2 subsections of the provided training and development data are cleaned up to produce plain text transcriptions where utterances are separated by a newline character. An iterative forced alignment script [14] which accepts plain text transcripts and ASR word lattice results as input, is used to compute the utterance level time alignments. Punctuation cleanup is also applied on the plain text transcripts in order to improve the accuracy of the alignment procedure, since ASR word lattice results do not involve any punctuation.

In each iteration, the alignment script uses confidence regions of the results of the previous iteration to narrow down the search space. The number of iterations is configured as three from previous experience.

2.1.2. Word level time alignment

The results from the forced alignment procedure described in the previous section have been used to generate stm type reference files to be used as the reference input to the NIST sclite ASR scoring utility. The ASR word lattice results are converted to ctm format which is accepted by the sclite utility as a hypothesis file with word alternative level time information. The NIST sclite utility was called with the “-o sgml” option in order to generate word level logs of correct words, substitutions, deletions and insertions as seen in Table 1. Information in these logs is used to find the word level timings of all the words in the reference files using linear time interpolation for the deletions. The computed word level timings are used to verify and update the utterance level start and end times computed with the forced alignment script described in the previous section.

<table>
<thead>
<tr>
<th>Type</th>
<th>Ref</th>
<th>Hyp</th>
<th>Reftime</th>
<th>Hyptime</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>españa</td>
<td>españa</td>
<td>10.640</td>
<td>10.700</td>
</tr>
<tr>
<td>I</td>
<td>un</td>
<td>entre</td>
<td>11.240</td>
<td>11.310</td>
</tr>
<tr>
<td>S</td>
<td>entra</td>
<td>entra</td>
<td>11.560</td>
<td>11.640</td>
</tr>
<tr>
<td>D</td>
<td>y</td>
<td>-</td>
<td>12.120</td>
<td>12.140</td>
</tr>
</tbody>
</table>

Word level timings can be obtained with the sclite utility without using the utterance level start and end times coming from the forced alignment procedure described in the previous section. However, the computation time in this case is on the order of days for a typical file from the training or development set and the results are not as accurate as the results obtained by the two-step process discussed in this work.

2.2. Data selection for training and testing

Utterance level time alignment information computed with the two-step procedure described in the previous section is used to convert the plain text transcripts into stm reference format accepted by NIST sclite utility where each utterance was labeled as a different speaker in order to enable utterance level ratios of correct, substitute, deleted and inserted words. The ASR best path results are used as the hypothesis input in ctm format. With the observation that the ratio of insertions is high for wrongly aligned utterances, the two criteria below are applied in order to select the utterances with correct alignments:

1) \( \% \) insertions < (\( \% \) correct + \( \% \) substitutions + \( \% \) deletions)
2) \( \% \) correct > 0

The provided ground truth transcriptions are assumed to be correct, and no analysis was carried out to test the correctness. 278 hours of training data is selected from train, dev1 and dev2 subsections of the provided data with the described data preparation and selection mechanism. 17 hours of data from dev2 subsection is reserved for testing and 261 hours of data from train and dev1 subsections are used for training the ASR model. The selected training data is combined with 91 hours of other European Spanish ASR training data from databases: Albayzin, Dihana, CORLEC-EHU, TC-STAR [5] to obtain 352 hours of training data for ASR task.

3. ASR model training

The Kaldi framework [11] was used to train the acoustic model of the ASR system submitted to Iberspeech 2018 Text to Speech Transcription Challenge by the Empathic team. Component diagram for ASR system training and testing is given in Figure 1.

3.1. Kaldi Framework

Kaldi is an open source toolkit for automatic speech recognition, intended for use by speech recognition researchers and professionals. It is composed of C++ binaries, utilities written in Bash Script, Python and Perl scripting languages and ready to run training and testing recipes with data preparation steps for various languages and scenarios.

3.2. Lexicon update

The vocabulary and the rule based phonetic descriptions of the base model is expanded using the out of vocabulary words detected in the new data selected from the provided training and development data. The final vocabulary size used in the ASR model training is 110124 words. However, for the testing with a language model a subset of this vocabulary is used.

3.3. Acoustic model building

The Kaldi Aspire recipe [15] which is the recipe used by the John Hopkin’s University team for the submission to the IARPA ASpIRE challenge was used for building the DNN-HMM acoustic model. High resolution MFCC’s with 40
coefficients and iVectors are used as input features. Gaussian posteriors used for the iVector estimation are based on the input features with Cepstral Mean and Variance Normalization (CMVN). GMM-HMM iterative phone alignment was used before starting the neural network training of the DNN-HMM training stage where mne3 chain implementation of the Kaldi framework was used with frame-subsampling-factor of 3, reducing number of output frames to 1/3 of the input frames. A 3-fold data augmentation is applied on the input acoustic features for the DNN-HMM training stage using the reverberation algorithm implemented in the Kaldi framework, by using noise databases RWCP, AIR and Reverb2014 in order to create multi-condition data of total 1057 hours.

A sub-sampled Time-delayed Deep Neural Network (TDNN) [16] with 6 layers and with ReLU and pnorm activation functions is used. Details of the neural network architecture and the stochastic gradient descent (SGD) based greedy layer-wise supervised training can be found in the system submission article for the Kaldi Aspire recipe [17].

TDNN training time for 2 epochs and 508 iterations is 13 hours 30 minutes with 3 NVidia GPUs (Quadro K6000, GeForce Titan X, GeForce Titan XP). Last iteration training and validation accuracies are 0.171631 and 0.199849 respectively. In order to avoid over-training, a selective system combination is carried out over all the iterations skipping the first 100 iterations, considering recorded accuracies for individual iteration results.

3.4. Language model (LM) adaptation

A 3-gram LM of the base model described in Section 2 is adapted using the selected training transcriptions of the provided data for the Iberspeech 2018 Speech to Text Transcription Challenge. Selected transcriptions are parsed with the ngram-count command of the SRILM utility up to order of 3–grams and calculated probabilities are mixed with the previous probabilities of the base model using ngram command with mixing coefficient 0.1. Vocabulary size for the produced LM is about 67000.

3.5. Model testing

The resulting ASR model is tested using selected audio and transcriptions from the dev2 subsection of the provided data for the Iberspeech 2018 training and development data. Data preparation and selection procedure described in Section 2 is used also for the preparation of the test data since the development data also suffers from the mis-alignment problem.

Audio files are segmented using voice activity detection based on adaptive thresholding method described by Otsu [18]. Feature extraction is applied on produced segments in order to obtain high resolution MFCC’s and iVectors with scaling factor 0.75. The extracted features are processed by a Viterbi beam decoder considering LM 3-gram probabilities coupled with TDNN tri-phone output probabilities. A GPU implementation of Kaldi mne3-latgen-faster decoder is used with parameters: --beam=15.0, --lattice-beam=2.5, max-active=7000, --acoustic-scale=1.0, --frame-subsampling-factor=2. Produced ASR lattice output is post-processed using minimum bayes risk (MBR) decoding [19] in order to find the best path result. ASR best path transcription results with word level timing information in ctm format are scored using the NIST sclite utility where ground truth transcriptions for selected utterances of dev2 subsection are used in stm reference format. Words from the ASR results which are not in the start end time interval of any reference utterance are omitted in the scoring process. An average WER result of 23.9 is obtained in the final model testing. Base model WER result obtained using the same testing method and decoding parameters is 35.3.

Real time factor in the decoding process including VAD, feature extraction and lattice post-processing is 0.022 using a 4 cores Intel i7-4820K CPU @ 3.70GHz and a single NVidia GeForce GTX 1080Ti GPU.

The provided test data for Iberspeech 2018 competition is processed with the produced ASR model using the same procedure described above and resulting transcriptions are submitted in plain text format.

![ASR system training / testing components](image)

Figure 1: ASR system training / testing components.

4. Discussion

The main challenge in the ASR model training and testing process with the provided data was the wrong or missing utterance level time alignment information of the provided training and development data. Therefore, time alignment and selection of the provided training and development data was necessary prior to acoustic model training. Since only selected utterances are used also in the model testing stage, and the ASR results which do not match with the selected time intervals are omitted from the WER result obtained by the model testing process using the dev2 subsection of the provided data, do not represent a WER result for all the data. If all the ground truth with true time alignments could be used, the obtained WER result is expected to be higher than is presented here since a selective process is used in the current WER calculation. However, the WER result calculated in the model testing
process was helpful for the benchmarking of the produced ASR model with the base model, and some other experiment results prior to the production of the final ASR model.

Much higher WER results are obtained (average WER 58.3 for the final produced model, average WER 105.7 for the base model) when they are calculated using all the provided ground truth text of the dev2 subsection ignoring provided wrong time information by using txt formatted ASR hypothesis files. A detailed analysis of the word level sclite logs shows that these values are not reliable because of mis-alignment problem of sclite utility usage without time information for such long reference and hypothesis files. This observation is the basis for the necessity of the two-step time alignment process used in this work prior to model training and testing.

A different value of frame-subsampling-factor compared to training process is chosen in the model testing (frame-subsampling-factor=3 in training and frame-subsampling-factor=2 in the testing) since it yields more accurate results in the testing of audio with Viterbi decoding using a LM.

5. Conclusion

The acoustic model building process with a data preparation and selection using a two-step time alignment procedure and utterance level thresholding with WER values yielded a good working acoustic model when the new training data is merged with the training data of the base model. The two-step time alignment procedure together with the utterance level data selection mechanism described in Section 2 enabled the usage of the provided data for the acoustic model training step of the ASR system generation. Model testing with the development data using an adapted version of the base LM showed a significant reduction in the WER results compared to the base model results used in the experiments.

6. Acknowledgements

The research leading to the results presented in this paper has been (partially) granted by the EU H2020 research and innovation program under grant number 769872.
7. References


GTM-UVIGO System for Albayzin 2018 Speech-to-Text Evaluation

Laura Docio-Fernandez, Carmen Garcia-Mateo
atlanTTic Research Center, Multimedia Technologies Group, University of Vigo, Spain
ldocio@gts.uvigo.es, carmen.garcia@uvigo.es

Abstract

This paper describes the Speech-to-Text system developed by the Multimedia Technologies Group (GTM) of the atlanTTic research center at the University of Vigo, for the Albayzin Speech-to-Text Challenge (S2T) organized in the Iberspeech 2018 conference. The large vocabulary automatic speech recognition system is built using the Kaldi toolkit. It uses an hybrid Deep Neural Network - Hidden Markov Model (DNN-HMM) for acoustic modeling, and a rescoring of a trigram based word-lattices, obtained in a first decoding stage, with a fourgram language model or a language model based on a recurrent neural network. The system was evaluated only on the open set training condition.

Index Terms: automatic speech recognition, deep neural networks, language model, speech activity detection

1. Introduction

Automatic Speech Recognition (ASR) is essential in many applications such as: dictation and transcription or captioning apps, speech-controlled interfaces, search engines for large multimedia archives, speech-to-speech translation, etc.

These days, with increases in computing power, the use of deep neural networks has spread to many fields including the ASR. With their use the performance of automatic speech recognition has been greatly improved. The current ASR systems are predominantly based on acoustic Hybrid Deep Neural Network–Hidden Markov Models (DNN-HMMs) [1] and the n-gram language model [2] [3]. However, in recent years, Neural Network Language Models (NNLM) have begun to be applied [4] [5] [6]. In these, words are embedded in a continuous space, in an attempt to map the semantic and grammatical information present in the training data, and in this way to achieve better generalization than n-gram models. The depth of the created network (the number of hidden layers), together with the ability to model a large number of context-dependent states, results in a reduction in Word Error Rate (WER). The type of neural networks most often used in language modelling are recurrent neural networks (RNNLMs). The recurrent connections present in these networks allow the modelling of long-range dependencies that improve the results obtained by n-gram models. In more recent work, recurrent network topologies such as LSTM (Long Short-Term Memory) [7] have also been applied [8] [9] [10].

In this paper, we present a summary of the GTM efforts in developing speech-to-text technology for Spanish language and its evaluation in the IberSPEECH-RTVE Speech to Text Transcription challenge. The submitted ASR system was evaluated on the open set training condition.

The paper is organized as it follows: in Section 2 the ASR system is described. Section 3 presents the data used to train the acoustic and language models. Section 4 describes the submitted systems specific characteristics, and finally Section 5 offers some final conclusions.

2. The GTM-UVIGO ASR system

This section describes the main blocks that comprises the ASR system.

2.1. Speech activity detection

The speech activity detection activity detection approach developed in the proposed system has four main stages. First, a voice activity detection (VAD) based on gaussian mixture models (GMMs) is applied to the audio signal in order to discard the non-speech intervals. Next, a two-step audio segmentation approach, based on the Bayesian information criterion (BIC) [11], is carried out. Once the audio segmentation output is obtained, those segments that are classified as music by a logistic regression classifier are discarded; this classifier relies on the i-vector paradigm for audio representation, as done in [12] [13].

The above stages use as acoustic features 19 Mel-frequency cepstral coefficients (MFCCs) plus energy, and a cepstral mean subtraction using a sliding window of 300 ms is applied.

2.2. Acoustic modeling

The acoustic models use a hybrid DNN-HMM modeling strategy with a neural network based on Dan Povey’s implementation in Kaldi [14]. This implementation uses a multi-spliced TDNN (Time Delay Neural Network) feed-forward architecture to model long-term temporal dependencies and short-term voice characteristics. The inputs to the network are 40 Mel-frequency cepstral coefficients extracted in the Feature Extraction block with a sampling frequency of 16 kHz. In each frame, we aggregate a 100-dimensional iVector to a 40-dimensional MFCC input.

The topology of this network consists of an input layer followed by 5 hidden layers with 1024 neurons with RELU activation function. Asymmetric input contexts were used, with more context in the background, which reduces the latency of the neuronal network in on-line decoding, and also because it seems to be more efficient from a WER perspective. Asymmetric contexts of 13 frames were used in the past, and 9 frames in the future. Figure 1 shows the topology used and Table 1 the layerwise context specification corresponding to this TDNN.

Table 1: Context specification of TDNN in Figure 1

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input context</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[-2, +2]</td>
</tr>
<tr>
<td>2</td>
<td>[-1, 2]</td>
</tr>
<tr>
<td>3</td>
<td>[-3, 3]</td>
</tr>
<tr>
<td>4</td>
<td>[-7, 2]</td>
</tr>
<tr>
<td>5</td>
<td>[0]</td>
</tr>
</tbody>
</table>
2.3. Language modeling

In terms of the language models, when working with n-gram LMs the model was trained using the SRI Language Modeling Toolkit. N-gram models of order 3 and 4 were used, that is, trigrams and fourgrams. A modified Kneser-Ney discounting of Chen and Goodman has also been applied, together with a weight interpolation with lower orders [15].

For training the RNNLMs, the Kaldi RNNLM [16] software was also used. The neural network language model is based on a RNN with 5 hidden layers and 800 neurons, where TDNN layers with activation function RELU, and LSTM layers are combined. The training is performed using Stochastic Gradient Descent (SGD), and in several epochs (in our case, 20 epochs). All RNNLM models have been trained with the same material as in the case of n-gram statistical models. Figure 2 shows the topology of the network used.

2.4. Recognition process

The recognition process is developed using the Kaldi toolkit [16]. The ASR system is based on two decoding stages to obtain the text transcription of the input speech signal. In the first decoding stage a lattice is obtained. This lattice is created using a 3-gram language model. In the second decoding stage, a language model rescoring is applied on this lattice. Figure 3 shows a block diagram of the recognition process.

3. Data resources

As stated in Section 1, the submitted ASR system was evaluated on the open set training condition. It uses acoustic models trained with data not in the RTVE2018DB, and language models trained with both text data in the RTVE2018DB and external text data.

Next, the data resources used for training the models are described.

3.1. Audio corpora

The data used for acoustic model training came from the following corpora:

- 2006 TC-STAR speech recognition evaluation campaign [17]: 79 hours of speaking in Spanish
- Galician broadcast news database Transcrigal [18]: 30 hours of speaking in Galician.

It must be noted that all the non-speech parts as well as the speech parts corresponding to transcriptions with pronunciation errors, incomplete sentences and short speech utterances were discarded, so in the end the acoustic training material consisted of approximately 109 hours (79 hours in Spanish and 30 hours in Galician).

3.2. Text corpora

The following text corpora were used to train the LMs:

- A text corpus of approximately 90M words composed of material from several sources: transcriptions of Eu-
european and Spanish Parliaments from the TC-STAR database, subtitles, books, newspapers, online courses and the transcriptions of the Mavir sessions included in the development set of the Albayzin 2016 Spoken Term Detection Evaluation. The vocabulary size of this corpus is approximately 250K words.

- The RTVE subtitles provided by the organizers of the evaluation. This text comprises approximately 60M words and its vocabulary size is approximately 173K words.

Table 2 shows the main characteristics of these text resources.

<table>
<thead>
<tr>
<th>Text Resource</th>
<th>Training text size</th>
<th>Vocabulary size</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC-STAR and Others</td>
<td>90M</td>
<td>250K</td>
</tr>
<tr>
<td>RTVE subtitles</td>
<td>60M</td>
<td>173K</td>
</tr>
</tbody>
</table>

From these text corpora, three LMs have been trained:

- **3-gram LM**: A trigram language model obtained by linear interpolation of two single trigram models. Each single model was trained using one of the above text resources.

- **4-gram LM**: A fourgram language model obtained by linear interpolation of two single fourgram models. Each single model was trained using one of the above text resources.

- **RNNLM**: A RNN language model trained with all the text described above.

The lexicon size of these models was approximately 312K words. The phonetic transcription of the lexicon words was automatically generated using the phonetic transcriber that forms part of the Cotovía GTM-UVIGO Text-to-Speech system [19].

### 4. Recognition results

This section presents the performance of the GTM-UVIGO ASR system on the development data provided by the organizers of the competition. Two systems that differ in the language model used in the decoding stage were evaluated. The primary system uses the 4-gram LM and the contrastive system uses the RNNLM, both described in Sections 2 and 3. The results obtained in the development set with the primary and contrastive systems are shown in Table 3. The table shows the average Word Error Rate (WER) by TV show and also the global average WER.

<table>
<thead>
<tr>
<th>System</th>
<th>Dev1</th>
<th>Dev2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>20H</td>
<td>12.13%</td>
</tr>
<tr>
<td></td>
<td>AP</td>
<td>22.82%</td>
</tr>
<tr>
<td></td>
<td>CA</td>
<td>53.95%</td>
</tr>
<tr>
<td></td>
<td>LM</td>
<td>30.81%</td>
</tr>
<tr>
<td></td>
<td>LN24H</td>
<td>27.22%</td>
</tr>
<tr>
<td></td>
<td>LN13H</td>
<td>27.85%</td>
</tr>
<tr>
<td>Average</td>
<td>26.65%</td>
<td>25.30%</td>
</tr>
<tr>
<td>Contrastive</td>
<td>Dev1</td>
<td>Dev2</td>
</tr>
<tr>
<td>Primary</td>
<td>24.30%</td>
<td>26.65%</td>
</tr>
<tr>
<td></td>
<td>Dev2</td>
<td>25.30%</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>24.78%</td>
</tr>
</tbody>
</table>

### 5. Conclusions and future work

In this paper, we have presented two ASR systems to IberSPEECH-RTVE Speech to Text Transcription challenge. The difference between these two systems lies in the language models used in the decoding stage. The primary system uses a 4-gram language model for rescoring and the contrastive system uses a language model based on recurrent neural networks.

As a future line of development we plan to improve the acoustic and language models of the ASR system, as well as to enrich their output with punctuation marks and correct capitalization.

---

1 MAVIR was a project funded by the Madrid region that coordinated several research groups and companies working on information retrieval (http://www.mavir.net)

### 6. Acknowledgements

This work has received financial support from the Spanish Ministerio de Economía y Competitividad through project 'TraceThem' (TEC2015-65345-P), from the Xunta de Galicia (Agrupación Estratexica Consolidada de Galicia accreditation 2016-2019) and the European Union (European Regional Development Fund ERDF).

### 7. References


Topic coherence analysis for the classification of Alzheimer’s disease

Anna Pompili\textsuperscript{1,2}, Alberto Abad\textsuperscript{1,2}, David Martins de Matos\textsuperscript{1,2}, Isabel Pavão Martins\textsuperscript{3,4}

\textsuperscript{1} L\textsuperscript{2}F – Spoken Language Systems Laboratory – INESC-ID Lisboa
\textsuperscript{2} Instituto Superior Técnico, Universidade de Lisboa, Portugal
\textsuperscript{3} Laboratório de Estudos de Linguagem, Faculty of Medicine, University of Lisbon, Portugal
\textsuperscript{4} Instituto de Medicina Molecular, Lisbon, Portugal
anna@l2f.inesc-id.pt

Abstract

Language impairment in Alzheimer’s disease is characterized by a decline in the semantic and pragmatic levels of language processing that manifests since the early stages of the disease. While semantic deficits have been widely investigated using linguistic features, pragmatic deficits are still mostly unexplored. In this work, we present an approach to automatically classify Alzheimer’s disease using a set of pragmatic features extracted from a discourse production task. Following the clinical practice, we consider an image representing a closed domain as a discourse’s elicitation form. Then, we model the elicited speech as a graph that encodes a hierarchy of topics. To do so, the proposed method relies on the integration of various NLP techniques: syntactic parsing for sentence segmentation into clauses, coreference resolution for capturing dependencies among clauses, and word embeddings for identifying semantic relations among topics. According to the experimental results, pragmatic features are able to provide promising results distinguishing individuals with Alzheimer’s disease, comparable to solutions based on other types of linguistic features.

Index Terms: natural language processing, topic coherence, Alzheimer’s disease

1. Introduction

In 2006 the prevalence of Alzheimer’s disease (AD) was 26.6 million people worldwide. Due to the increase of average lifespan, it is expected that this data will quadruple by 2050, affecting 1 in 85 persons worldwide \cite{1}. No treatments stop or reverse the progression of the disease, though some may temporarily improve the symptoms. AD is currently diagnosed through an analysis of the patient history and through neuropsychological tests assessing cognitive decline in different domains (memory, reasoning, language, and visuospatial abilities). In fact, although the prominent symptom of the disease is memory impairment, language problems are also prevalent and existing literature confirms they are an important factor \cite{2, 3, 4, 5, 6}. Impairments in language abilities are usually the result of a decline either in the semantic or pragmatic levels of language processing. Semantic processing is related with the content of language and involve words and their meaning. Deficits in this domain are typically characterized by difficulties in word finding, word comprehension, semantic paraphasia, and by the use of a reduced vocabulary. Pragmatic processing, on the other hand, is concerned with the inappropriate use of language in social situations. Deficits in this domain may include difficulties in understanding questions, in following conversations, and in identifying the key points of a story, getting lost in the details. It is likely that the semantic and pragmatic levels are interdependent, and that semantic deficits in word finding may contribute to pragmatic deficits that lead, for example, to the problem of maintaining the topic of a conversation \cite{7}.

In the recent years, there has been a growing interest from the research community in the computational analysis of language impairment in AD. Overall, existing studies targeted the automatic assessment of lexical, syntactical, and semantic deficits through an extensive amount of linguistic features \cite{8, 9, 10}. More recently, semantic changes have been also investigated through vector space model representations \cite{11, 12}. On the other hand, up to our knowledge, there are no works facing language impairments at an higher level of processing, considering macro-linguistic aspects of discourse production such as cohesion and coherence. While cohesion expresses the semantic relationship between elements, coherence is related to the conceptual organization of speech, and may be analyzed through the study of local, global, and topic coherence.

In this work, we investigate the possibility of automatically discriminate AD exploring a novel approach, based on the analysis of topic coherence. To this end, we model discourse transcripts into graphs encoding a hierarchy of topics on which we compute a relatively small set of pragmatic features. In the following, in Section 2, topic coherence analysis is briefly introduced, followed by an overview of the current state of the art for AD classification. Then, in Section 3 and 4, we present the dataset used in this study and a description of our methodology. Finally, the features related with topic coherence and classification results are reported in Section 5 and 6, respectively. Conclusions are summarized in Section 7.

2. Related work

2.1. Introduction to topic coherence analysis

The notion of topic and subtopic was introduced in 1991 in the work of Mentis & Prutting \cite{13}, whose target was an analysis of topic introduction and maintenance. A topic was defined as a clause identifying the question of immediate concern, while a subtopic being an elaboration or expansion of one aspect of the main topic.

Several years later, Bradie \textit{et al.} \cite{14} analyzed topic coherence and topic maintenance in individuals with right hemisphere brain damage. This work extends the one of Mentis & Prutting \cite{13} with the inclusion of the notion of sub-subtopic and sub-sub-subtopic. Topic and sub-divisional structures were further categorized as new, related, or reintroduced.

In a following study Mackenzie \textit{et al.} \cite{15} used discourse samples elicited through a picture description task to determine the influences of age, education, and gender on the concepts and topic coherence of 225 healthy adults. Results confirmed education level as a highly important variable affecting the per-
formance of healthy adults.

More recently, Miranda [16] investigated the influence of education in the macro-linguistic dimension of discourse evaluation, considering concepts analysis, local, global and topic coherence, and cohesion. The study was performed on a population of 87 healthy, elderly Portuguese participants. Results corroborated the ones obtained by Mackenzie et al. [15], confirming the effect of literacy in this type of analysis.

2.2. Computational work for AD classification

From a computational point of view, language impairment in AD has been extensively assessed through the analysis of linguistic and acoustic features. Among the various works, we mention the study of Fraser et al. [8], where the authors considered more than 350 features to capture lexical, syntactic, grammatical, and semantic phenomena. Using a selection of 35 features the authors achieved state of the art classification accuracy of over 81% in distinguishing individuals with AD.

Yancheva et al. [11] presented a generalizable method to automatically generate and evaluate the information content conveyed from the description of the Cookie Theft picture. The authors created two cluster models, one for each group, from which they extracted different semantic features. Classification accuracy results achieved an F-score of 0.74. By combining semantic features to the set of lexicosyntactic features used in the work of Fraser et al. [8] the F-score improves to 0.80.

To the extent of our knowledge, the first work approaching coherence and cohesion computationally is the study of dos Santos et al. [17], although the target of the authors is the detection of Mild Cognitive Impairment (MCI). To this purpose, they model discourse transcripts with a complex network enriched with word embeddings. Classification is performed using topological metrics of the network and linguistic features, among which referential cohesion. Accuracy varies among 52%, 65%, and 74%, depending on the dataset used.

3. The Cookie Theft corpus

Data used in this work are obtained from the DementiaBank database*, which is part of the larger TalkBank project [18, 19]. The collection was gathered in the context of a yearly basis, longitudinal study: demographics data, together with the education level, are provided. Participants included elderly controls, people with MCI and different type of Dementia. Among other assessments, participants were required to provide the description of the Cookie Theft picture, shown in Figure 1. Each speech sample was recorded and then manually transcribed at the word level following the TalkBank CHAT (Codes for the Human Analysis of Transcripts) protocol [20]. Data are in English language.

For the purposes of this study, only participants with a diagnosis of AD were selected, resulting in 234 speech samples from 147 patients. Control participants were also included, resulting in 241 speech samples from 98 speakers.

4. Modeling discourse transcripts as a hierarchy of topics

The topics used during discourse production may be subject to an internal, structural organization in order to achieve an information hierarchy. This organizational structure allows a gradual organization of information that is essential for an effective communication [22]. In fact, being important for both the speaker and the listener, this type of organization highlights the key concepts and indicates the degrees of importance and relevance within the discourse. Mackenzie et al. [15] in their work provided an example of a topic hierarchy based on the Cookie Theft picture description task, which was later extended in the study of Miranda [16]. To better understand the problem at hand, an excerpt of this hierarchy is also reported in Figure 2.

The number of topics that can be described observing the Cookie Theft picture, is somehow limited to the concepts that are explicitly represented in the scene, and to those ones that can be inferred from the previous (e.g., climate). Taking this into account, the problem of building a topic hierarchy from a transcript can be modeled with a semi-supervised approach in which a predefined set of topics clusters is used to guide the assignment of a new topic to a level in the hierarchy.

Both for the creation of the topics clusters, and for the analysis of a new discourse sample, a multistage approach is used to prepare, enhance, and transform the original transcriptions in a representation suitable for the subsequent analysis. Initially the transcriptions are preprocessed, then syntactical information is used to separate sentences into clauses and to identify coreferential expressions. Finally, we compute the vector representation of each clause by averaging the embeddings extracted for each word. To build the graph representing the topic hierarchy we develop an algorithm based on the cosine similarity that first evaluates the membership of a clause to the topic clusters, and then assigns the clause to a node in the hierarchy. We account for new and repeated topics. Each stage of this process is better described in the following sections.

4.1. Preprocessing

The Cookie Theft corpus provides the textual transcriptions of the participant’s speech samples together with the morphological analysis and an extensive set of manual annotations (i.e., disfluencies, pauses, repetitions, and other more complex events). Among these, retracing and reformulation are used to indicate abandoned sentences where the speaker starts to say something, but then stops. While in the former the speaker may maintain the same idea changing the syntax, the latter involves a complete restatement of the idea. In order to prepare the transcriptions for the next stage of the pipeline, all the annotations were removed, and in the case of a retrace or a reformulation, also

* https://dementia.talkbank.org
Figure 2: An excerpt of a topic hierarchy for the Cookie Theft picture found in the work of Miranda [16] (text was translated from Portuguese).

The text marked as being substituted was ignored. Additionally, disfluencies were disregarded and contractions were expanded to their canonical form. At this stage of processing, stopwords were not removed. Once the preprocessing phase is concluded, Part of Speech (POS) tags are automatically generated using the lexicalized probabilistic parser of the Stanford University [23].

4.2. Clause segmentation

The next step requires to face the problem of identifying dependent and independent clauses. In fact, while the Cookie Theft corpus already provides a segmentation of the input speech into sentences, this is not sufficient for the purposes of this work. Complex, compound or complex-compound sentences may contain references to multiple topics. The following excerpt shows an example of this problem, a complex sentence may contain references to multiple topics. The following excerpt shows an example of this problem, a complex sentence composed of a dependent and an independent clause: / and the mother is washing dishes while the water is running over in the sink on the floor."

A possible way to cope with the separation of different sentences types is by using syntactic parse trees. Thus, in a similar way to the work of Feng et al. [24], POS tags are used for the identification of dependent and independent clauses. For the former, the tag SBAR is used, while for the latter, the proposed solution checks the sequence of nodes along the tree to verify if the tag S or the tags [NP VP] appear in the sequence.

4.3. Coreference analysis

The analysis of coreference proves to be particularly useful in higher level NLP applications that involve language understanding, such as an extended discourse [25]. Strictly related with the notions of anaphora and cataphora, coreference resolution goes beyond the relation of dependence implicated by these concepts. It allows to identify when two or more expressions refer to the same entity in a text. In this work, the analysis of coreference has been performed with the Stanford coreference resolution system [26], using the results of the segmentation performed in the previous step. During the process of building the hierarchy, the coreference information is used to guide the assignment of a subtopic to the corresponding level in the hierarchy. To this purpose, we constrain the results provided by the coreference system to those mentions whose referent is the subject of the sentence. We are not interested in considering other coreferential expressions because a subtopic, being a specialization of a topic, is typically referred to the subject of the sentence.

4.4. Sentence embeddings

In the last step of the pipeline, discourse transcripts are transformed in a representation suitable to compare and measure differences between sentences. In particular, the transformed transcripts should be robust to syntactic and lexical differences and should provide the capability to capture semantic regularities among sentences. To this purpose, we rely on a pre-trained model of word vector representations containing 2 million word vectors, in 300 dimensions, trained with fastText on Common Crawl [27]. In the process of converting a sentence into its vector space representation, we first perform a selection of four lexical items (nouns, pronouns, verb, adjectives), then, for each word we extract the corresponding word vector and finally we compute the average over the whole sentence.

4.5. Topic hierarchy analysis

To create a topic hierarchy from a transcript, we follow a methodology that is partly inspired by current clinical practice. Thus, in modelling the problem we do not want to impose a predefined order or structure in the way topics and subtopics may be presented, as this, of course, will depend on how the discourse is organized. However, we can take advantage of the closed domain nature of the task to define a reduced number of clusters of broad topics that will help to guide the construction of the hierarchy and the identification of off-topic clauses.

4.5.1. Topic clusters definition

As mentioned, the proposed solution relies on the supervised creation of a predefined number of clusters of broad topics. Each cluster contains a representative set of sentences that are related with the topic of the cluster. 10 clusters were defined: main scene, mother, boy, girl, children, garden, climate, not-related, incomplete, and no-content. The cluster not-related was used to model those sentences in which the participant is not performing the task (e.g., questions directed to the interviewer). The clusters incomplete and no-content are instead used to explicitly model sentences that may be characteristics of a language impairment. The former contains fragments of text that do not represent a complete sentence (e.g., /overflowing sink/), the latter identifies those expressions that do not add semantic information about the image (e.g., /fortunately there is nothing happening out there/ /what is going on/). To build the clusters, 30% of the data from both the AD and the control group is used. Each sentence has been manually annotated with the corresponding cluster label and clusters are simply modelled by the complete set of sentences belonging to them.

4.5.2. Topic hierarchy building algorithm

The algorithm to build the topic hierarchy relies on the cosine similarity between sentence embeddings. The first step consists in verifying to which topic cluster belongs the current sentence. This is achieved by computing the cosine similarity between the current sentence embeddings and each sentence embeddings in

---

http://www.surdecanu.info/mihai/teaching/istat555-fall13/readings/PennTreebankConstituents.html
each topic clusters. The highest result determines the cluster for the new sentence. In the following step, we need to assign the current sentence embeddings to a level in the current hierarchy. This implies to establish if we are dealing with a new or a repeated topic and its level of specialization (i.e., subtopic, sub-subtopic, etc.). This is achieved by first identifying, in the current hierarchy, the sub-graph whose nodes belong to the same cluster of the current sentence (e.g., the sub-graph corresponding to the mother cluster). Then, we compute the cosine similarity between the current sentence and each nodes of this sub-graph. The new sentence is considered a son of the closest node if the similarity is lower than a threshold. Otherwise, it is considered a repeated topic. If there is no sub-graph, the sentence embedding is added as a new topic. If the new topic results to be a coreferential expression, this kind of information supersedes the cosine metric strategy, and the new topic is added directly as a son of its referent.

Although the algorithm developed resembles the analysis performed in the standard clinical practice, the aim of this work is not the comparison of the automatic method with the manual one. Instead, our focus is understanding if pragmatic features related with topic coherence analysis may be relevant to discriminate AD. The type of features computed, as well as the results of classification experiments are described in the following sections.

5. Topic coherence features

Through the multistage approach and the final hierarchy of topics we identified sixteen measurements: (1-4) the number of topics, subtopics, sub-subtopics and sub-sub-subtopics introduced, (5-6) the proportion of dependent and independent clauses to the total number of sentences, (7) the total number of coreferential mentions, (8) the total number of topics, subtopics, sub-subtopics and sub-sub-subtopics repeated, (9-11) the number of sentences that were classified as not-related, incomplete, or no-content in the first step of the main algorithm, (12) the coefficient of variation (the ratio of the standard deviation to the mean) of the cosine similarity between two temporally consecutive topics, (13) the length of the longest path from the root node to all leaves, (14) the average number of outgoing edges of all nodes, (15) the total number of sentences, (16) the ratio of dependent to independent clauses.

6. Results and discussion

Classification experiments were performed with a Random Forest classifier, using the 70% of the remaining data of the Cookie Theft corpus, once that 30% of the data was retained to model the topic hierarchy. A stratified k-fold cross validation per subject strategy was implemented, with k being equal to 10.

Initial results, using the set of features described previously, provided an average accuracy of 74% in distinguishing AD patients from healthy controls. Then, in order to understand the importance of each feature, we implemented a forward feature selection method. This is an iterative approach in which the model is trained with a varying number of features. Starting with no features, at each iteration we test the accuracy of the model by adding, one at a time, each of the features that were not selected in a previous iteration. The accuracy is evaluated with a stratified 10-fold cross validation. The feature that yields the best accuracy is retained for further processing. The results of this method are shown in Figure 3. With this approach, we identified the first six features as the most relevant in discriminating the disease.

Our results, achieved with the 70% of the data, provided an average accuracy of 77%, and an average F-score of 77% in classifying AD. Interestingly, the number of topics was the first feature selected, providing, alone, an average accuracy of 67%. Comparing these results with current state of the art, we acknowledge that Fraser et al. [8] achieved a higher accuracy (81%) using a set of lexicosyntactic features. On the other hand, we also recognized that these results are slightly better than the ones achieved by Yancheva et al. [11] (F-score 74%) using only a set of 12 semantic features. However, when the authors combine lexicosyntactic and semantic information, the F-score improves to 80%. These considerations are interesting for multiple reasons, in fact, on one side they confirm the relevance of pragmatic features related with topic coherence in the task of classifying AD. On the other hand, they also highlight that lexicosyntactic features are extremely important in characterizing the disease and should be used in a complementary way with other features.

7. Conclusions

In this work, we approached the problem of exploiting topic coherence analysis to automatically classify AD. To this purpose, we proposed an algorithm inspired by the type of assessment conducted by clinicians to construct the topic hierarchy of a picture description task, from which we extract a reduced set of pragmatic features for automatic classification. Initial experimental results show comparable AD classification performance to current state of the art approaches using different types of consolidated linguistic features. As future work, we plan to integrate the proposed pragmatic features with lexicosyntactic features and to explore the extension of this kind of analysis to other types of discourse production tasks, including open-domain tasks.

8. Acknowledgments

The authors want to express their gratitude to Dr. Filipa Miranda for her precious advice and help provided during the development of this work. This work was supported by Portuguese national funds through – Fundação para a Ciência e a Tecnologia (FCT), under Grants SFRH/BD/97187/2013 and Project with reference UID/CEC/50021/2013.
9. References


Building a global dictionary for semantic technologies

Eszter Iklódi\textsuperscript{1}, Gábor Recski\textsuperscript{1}, Gábor Borbély\textsuperscript{2}, Maria Jose Castro-Bleda\textsuperscript{3}

\textsuperscript{1}Department of Automation and Applied Informatics, Budapest University of Technology and Economics, Budapest, Hungary
\textsuperscript{2}Department of Algebra, Budapest University of Technology and Economics, Budapest, Hungary
\textsuperscript{3}Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, Valencia, Spain

\email{eszter.iklodi@gmail.com, recski.gabor@aut.bme.hu, borbely@math.bme.hu, mcastro@dsic.upv.es}

Abstract

This paper proposes a novel method for finding linear mappings among word vectors for various languages. Compared to previous approaches, this method does not learn translation matrices between two specific languages, but between a given language and a shared, universal space. The system was trained in two different modes, first between two languages, and after that applying three languages at the same time. In the first case two different training data were applied; Dinu’s English-Italian benchmark data \cite{dinu2017}, and English-Italian translation pairs extracted from the PanLex database \cite{mcastro2016}. In the second case only the PanLex database was used.

The system performs on English-Italian languages with the best setting significantly better than the baseline system of Mikolov et al. \cite{mikolov2013}, and it provides a comparable performance with the more sophisticated systems of Faruqui and Dyer \cite{faruqui2016} and Dinu et al. \cite{dinu2017}. Exploiting the richness of the PanLex database, the proposed method makes it possible to learn linear mappings among an arbitrary number of languages.

Index Terms: semantics, word embeddings, multilingual embeddings, translation, artificial neural networks

1. Introduction

Computer-driven natural language processing plays an increasingly important role in our everyday life. In the current digital world, using natural language for human-machine communication has become a basic requirement. In order to meet this requirement, it is inevitable to analyze human languages semantically. Nowadays, state-of-the-art systems represent word meaning with high dimensional vectors, known as word embeddings.

Current embedding models are learned from monolingual corpora, and therefore infer language dependency. But one might ask if the structure of the different embeddings, i.e. different meaning representations, are universal among all human languages. Youn et al. \cite{youn2018} proposed a procedure for building graphs from concepts of different languages. They found that these graphs reflected a certain structure of meaning with respect to the languages they were built of. They concluded that the structural properties of these graphs are consistent across different language groups, and largely independent of geography, environment, and the presence or absence of literary traditions. Such findings led to a new research direction within the field of computational semantics, which focuses on the construction of universal meaning representations, most of the times in the form of cross-lingual word embedding models \cite{kim2014}.

One way to create such models is to find mappings between embeddings of different languages \cite{kim2014, kim2015, kim2016}. Our work proposes a novel procedure for learning such mappings in the form of translation matrices that serve to map each language to a universal space.

Section 2 summarizes the progress made on learning translation matrices between word embeddings over the last five years. Section 3 discusses the proposed method in detail. Following that, Section 4 describes the experimental setup we used and reports the obtained results. Finally, Section 5 concludes the advantages and disadvantages of the proposed model, and also discusses some improvements for future work.

2. Related work

In 2013, Mikolov et al. \cite{mikolov2013} published a simple two-step procedure for creating universal embeddings. In the first step they built monolingual models of languages using huge corpora, and in the second step a small bilingual dictionary was used to learn linear projection between the languages. The optimization problem was the following:

\begin{equation}
\min_W \sum_{i=1}^n \|Wx_i - z_i\|^2
\end{equation}

where $W$ denotes the transformation matrix, and $\{x_i, z_i\}_{i=1}^n$ are the continuous vector representations of word translation pairs, with $x_i$ being in the source language space and $z_i$ in the target language space.

Faruqui and Dyer \cite{faruqui2016} proposed a procedure to obtain multilingual word embeddings by concatenating the two word vectors coming from the two languages, applying canonical correlation analysis (CCA). Xing et al. \cite{xing2014} found that bilingual translation can be largely improved by normalizing the embeddings and by restricting the transformation matrices into orthogonal ones. Dinu et al. \cite{dinu2017} showed that the neighbourhoods of the mapped vectors are strongly polluted by hubs, which are vectors that tend to be near a high proportion of items. They proposed a method that computes hubness scores for target space vectors and penalizes those vectors that are close to many words, i.e. hubs are down-ranked in the neighbouring lists. Lazaridou et al. \cite{lazaridou2016} studied some theoretical and empirical properties of a general cross-space mapping function, and tested them on cross-linguistic (word translation) and cross-modal (image labelling) tasks. They also introduced the use of negative samples during the learning process. Amar et al. \cite{amar2019} proposed methods for estimating and evaluating embeddings of words in more than fifty languages in a single shared embedding space. Since
English usually offers the largest corpora and bilingual dictionaries, they used the English embeddings to serve as the shared embedding space. Artetxe et al. [12] built a generic framework that generalizes previous works made on cross-linguistic embeddings and they concluded that the best systems were the ones with orthogonality constraint and a global pre-processing with length normalization and dimension-wise mean centering. Smith et al. [7] also proved that translation matrices should be orthogonal, for which they applied singular value decomposition (SVD) on the transformation matrices. Besides, they also introduced a novel “inverted softmax” method for identifying translation pairs. All these works listed above applied supervised learning. However, in 2017 Conneau et al. [8] introduced an unsupervised way for aligning monolingual word embedding spaces between two languages without using any parallel corpora. This unsupervised procedure holds the current state-of-the-art results on Dinuz’s benchmark word translation task. For comparing the different results see Table 1 and Table 2.

3. Proposed method

We propose a method that learns linear mappings between word translation pairs in the form of translation matrices that map pre-trained word embeddings into a universal vector space. During training, the cosine similarity of word translation pairs is maximized, which is calculated in the universal space. The method is applicable for any number of languages. Since, independent of the number of languages applied during training, for each language always exactly one translation matrix is learned, by introducing new languages, the number of the learned parameters remains linear to the number of the applied languages.

Let $L$ be a set of languages, and $TP$ a set of translation pairs where each entry is a tuple of two in the form of $(w_1, w_2)$ where $w_1$ is a word in language $L_1$ and $w_2$ is a word in language $L_2$, and both $L_1$ and $L_2$ are in $L$. Then, let’s consider the following equation to optimize:

$$\frac{1}{|TP|} \sum_{L_1 \in L, \ L_2 \in L} \sum_{(w_1, w_2) \in TP} \cos_sim(w_1 \cdot T_1, w_2 \cdot T_2)$$

(2)

where $T_1$ and $T_2$ are translation matrices mapping $L_1$ and $L_2$ to the universal space. Since the equation is normalized with the number of translation pairs in the $TP$ set, the optimal value of this function is 1. Off-the-shelf optimizers are programmed to find local minimum values, so during the training process the loss function is multiplied by $-1$. Word vectors are always normalized, so the $cos_sim$ reduces to a simple dot product.

At test time, first, both source and target language words are mapped into the universal space, and from the most frequent 200k mapped target language words a look-up space is defined. Then, the system is evaluated with the Precision metric, more specifically with Precision @1, @5, and @10, where Precision @N denotes the percentage of how many times the real translation of a source word is found among the N closest word vectors in the look-up space. The distance assigned to the word vectors when searching in the look-up space is the $cos_sim$.

Previous works, such as Mikolov et al. [7] or Conneau et al. [8], suggested restricting the transformation matrix to an orthogonal one. From an arbitrary transformation matrix $T$ an orthogonal $T'$ can be obtained by applying the SVD procedure. Our experiments showed that by applying SVD on the transformation matrices the learning is significantly faster. Best results were obtained when applying the SVD only once, at the beginning of the learning process.

4. Experiments

4.1. Experimental setup

4.1.1. Pre-trained word embeddings

For pre-trained word embeddings we took the fastText embeddings published by Conneau et al. [8]. These embeddings were trained by applying their novel method where words are represented as a bag of character n-grams [13]. This model outperformed Mikolov’s [14] CBOW and skipgram baseline systems that did not take any sub-word information into account. Conneau’s pre-trained word vectors trained on Wikipedia are available for 294 languages.

Some experiments were also run by using the same embedding that was used by Dinuz et al. [1] in their experiments. These word vectors were trained with word2vec and then they were mapped onto 300 dimensional vectors. The most common words in both the English and Italian corpora were extracted. The English word vectors were trained on the WackyPedia/ukWaC and BNC corpora, while the Italian word vectors were trained on the WackyPedia/itWaC corpus. This word embedding will be referred to as the WaCky embedding.

4.1.2. Gold dictionaries

First, we ran the experiments on Dinuz’s English-Italian benchmark data [1]. It is an English-Italian gold dictionary split into a train and a test set, which was built from Europarl en-it [15]. For the test set they used 1,500 English words split into 5 frequency bins, 300 randomly chosen in each bin. The bins are defined in terms of rank in the frequency-sorted lexicon: [1-5K], [5K-20K], [20K-50K], [50K-100K], and [100K-200K]. Some of these 1500 English words have multiple Italian translations in the Europarl dictionary, so the resulting test set contains 1869 word pairs all together, with 1500 different English, and with 1849 different Italian words. For the training set the top 5k entries were extracted and care was taken to avoid any overlap with test elements on the English side. On the Italian side, however, an overlap of 113 words is still present. In the end the train set contains 5k word pairs with 3442 different English, and 4549 different Italian words.

Then, we built another golden dictionary similar to that of Dinuz’s, but this time the translation pairs were extracted from the PanLex [2] database. PanLex is a nonprofit organization that aims to build a multilingual lexical database from available dictionaries made by domain experts in all languages. To each translation pair a confidence value is assigned, which can be used for filtering the extracted data. These confidence values are in the range of [1, 9], with 9 meaning high and 1 meaning low confidence. During the extraction process, translations with a confidence value below 7 and those for which no word vector was found in the fastText embedding were dropped. Then, training and test sets were constructed following Dinuz’s steps, except for that only those English words were taken for which only one Italian translation was present. Experiments showed that otherwise a serious noise was brought into the system, since in many cases one English word might have up to 10 different Italian translations.

---

1 http://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md
2 http://opus.lingfil.uu.se/
4.2. Results

4.2.1. Parameter adjustment using Dinu's data

First, parameter adjustment was performed using Dinu’s data, which gave 0.1 as the best learning rate and 64 as the best batch size, where batch size is equal to the number of translation pairs used in one iteration. With applying SVD only once at the beginning the obtained results of our best system are significantly worse than state-of-the-art results on this benchmark data, but they are comparable with or even better than some of the previous models discussed in Section 2. For comparison see Table 1 and Table 2.

<table>
<thead>
<tr>
<th></th>
<th>@1</th>
<th>@5</th>
<th>@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mikolov et al.</td>
<td>0.338</td>
<td>0.483</td>
<td>0.539</td>
</tr>
<tr>
<td>Faruqui et al.</td>
<td>0.361</td>
<td>0.527</td>
<td>0.581</td>
</tr>
<tr>
<td>Dinu et al.</td>
<td>0.385</td>
<td>0.564</td>
<td>0.639</td>
</tr>
<tr>
<td>Smith et al. (2017)</td>
<td>0.431</td>
<td>0.607</td>
<td>0.651</td>
</tr>
<tr>
<td>Conneau et al. (2017) - WaCky</td>
<td>0.451</td>
<td>0.607</td>
<td>0.651</td>
</tr>
<tr>
<td>Conneau et al. (2017) - fastText</td>
<td><strong>0.662</strong></td>
<td><strong>0.804</strong></td>
<td><strong>0.834</strong></td>
</tr>
</tbody>
</table>

The obtained results show that training on the PanLex data cannot beat the system trained on Dinu’s data, which performs better both on Dinu’s and on the PanLex test sets. Not even combining the two training sets succeeds in achieving significantly better results, although on the PanLex test set it does improve the scores in the Italian-English direction.

4.2.4. Continuing the training with PanLex data

Another experiment was conducted to continue the baseline system trained on Dinu’s data with the PanLex data. In other words, it is the same as initializing the translation matrices of the PanLex training process with previously learned ones. The baseline system reaches its best performance between 2000 and 4000 epochs, depending on which precision value is regarded. Table 7 shows that on the English-Italian task there is no improvement at all, while on the Italian-English task with the best setting slightly better scores are achieved on precision @1 and @10 values.

4.2.5. Experiments using three languages

Finally, a multilingual experiment was carried out where the system was trained on three languages - English, Italian, and Spanish - at the same time. During training the system learns three different translation matrices, one for English-universal, one for Italian-universal, and one for Spanish-universal space mapping. For example, in order to learn the English-universal translation matrix, both the English-Italian and the English-Spanish dictionaries are used, according to Equation (2). Batches are homogeneous, but two following batches are always different in terms of the language origins of the contained data. That is, first an English-Italian batch is fed to the system, then an English-Spanish batch, after that an Italian-Spanish batch, and so on. First, bilingual models were trained in order to compare them later with the multilingual system. The results of the bilingual models are summarized in Table 8. Results are best on the Italian-Spanish task. Next, the system was trained using all the three languages at the same time. During the training process the model was evaluated on the bilingual test datasets of which the results are shown in Table 9. The obtained results show that no advantage was achieved by extending the number of languages, since the multilingual model performs worse than any of the pairwise bilingual models.

5. Conclusions and future work

This paper proposes a novel method for finding linear mappings between word embeddings in different languages. As a proof of concept a framework was developed which enabled basic parameter adjustments and flexible configuration for initial experimentation.
### Table 5. Comparing Dinu’s and PanLex data on Dinu’s test set

<table>
<thead>
<tr>
<th></th>
<th>eng-ita</th>
<th>ita-eng</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@1</td>
<td>@5</td>
<td>@10</td>
<td>@1</td>
<td>@5</td>
</tr>
<tr>
<td>train:Dinu - test:old</td>
<td>0.3770</td>
<td>0.5647</td>
<td>0.6245</td>
<td>0.3103</td>
<td>0.5018</td>
</tr>
<tr>
<td></td>
<td>0.3560</td>
<td>0.5407</td>
<td>0.5978</td>
<td>0.2917</td>
<td>0.4792</td>
</tr>
<tr>
<td>train:PanLex - test:new</td>
<td>0.1360</td>
<td>0.2309</td>
<td>0.2594</td>
<td>0.1361</td>
<td>0.2556</td>
</tr>
<tr>
<td>train:Dinu+PanLex - test:new</td>
<td>0.2930</td>
<td>0.4349</td>
<td>0.4861</td>
<td>0.2910</td>
<td>0.4556</td>
</tr>
</tbody>
</table>

### Table 6. Comparing Dinu’s and PanLex data on the PanLex test set

<table>
<thead>
<tr>
<th></th>
<th>eng-ita</th>
<th>ita-eng</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@1</td>
<td>@5</td>
<td>@10</td>
<td>@1</td>
<td>@5</td>
</tr>
<tr>
<td>train:PanLex - test:old</td>
<td>0.1960</td>
<td>0.3087</td>
<td>0.3440</td>
<td>0.1838</td>
<td>0.3059</td>
</tr>
<tr>
<td></td>
<td>0.1812</td>
<td>0.2858</td>
<td>0.3196</td>
<td>0.1668</td>
<td>0.2835</td>
</tr>
<tr>
<td>train:Dinu - test:new</td>
<td>0.2295</td>
<td>0.4171</td>
<td>0.4839</td>
<td>0.2227</td>
<td>0.3763</td>
</tr>
<tr>
<td></td>
<td>0.2295</td>
<td>0.3712</td>
<td>0.4275</td>
<td>0.2498</td>
<td>0.4026</td>
</tr>
</tbody>
</table>

### Table 7. Continuing the baseline system with the PanLex data.

<table>
<thead>
<tr>
<th></th>
<th>eng-ita</th>
<th>ita-eng</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@1</td>
<td>@5</td>
<td>@10</td>
<td>@1</td>
<td>@5</td>
</tr>
<tr>
<td>original</td>
<td>0.3770</td>
<td>0.5647</td>
<td>0.6245</td>
<td>0.3103</td>
<td>0.5018</td>
</tr>
<tr>
<td>cont from 2000</td>
<td>0.3426</td>
<td>0.5256</td>
<td>0.5802</td>
<td>0.3229</td>
<td>0.4882</td>
</tr>
<tr>
<td>cont from 3000</td>
<td>0.3535</td>
<td>0.5416</td>
<td>0.5970</td>
<td>0.3229</td>
<td>0.4840</td>
</tr>
<tr>
<td>cont from 4000</td>
<td>0.3510</td>
<td>0.5273</td>
<td>0.5911</td>
<td>0.3118</td>
<td>0.4701</td>
</tr>
</tbody>
</table>

### Table 8. Results of bilingual models trained pairwise on the three different languages.

<table>
<thead>
<tr>
<th></th>
<th>L1-L2</th>
<th>L2-L1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@1</td>
<td>@5</td>
</tr>
<tr>
<td>eng-ita</td>
<td>0.2080</td>
<td>0.3280</td>
</tr>
<tr>
<td>eng-spa</td>
<td>0.2840</td>
<td>0.4320</td>
</tr>
<tr>
<td>spa-ita</td>
<td>0.3920</td>
<td>0.5340</td>
</tr>
</tbody>
</table>

### Table 9. Bilingual results of the multilingual model trained using three different languages at the same time.

<table>
<thead>
<tr>
<th></th>
<th>L1-L2</th>
<th>L2-L1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@1</td>
<td>@5</td>
</tr>
<tr>
<td>eng-ita</td>
<td>0.1573</td>
<td>0.2667</td>
</tr>
<tr>
<td>eng-spa</td>
<td>0.1947</td>
<td>0.2973</td>
</tr>
<tr>
<td>spa-ita</td>
<td>0.2520</td>
<td>0.3640</td>
</tr>
</tbody>
</table>

An interesting finding was that the system learned much faster when an initial SVD was applied on the translation matrices. Results obtained with these settings on Dinu’s data showed that the proposed model did learn from the data. The obtained precision scores, though, were far from current state-of-the-art results on this benchmark data, they were comparable with results of previous attempts. The proposed model performed much better using the fastText embeddings [8], than using Dinu’s WaCky embeddings [1].

Thereafter, an English-Italian dataset was extracted from the PanLex database, from which training and test datasets were constructed roughly following the same steps that Dinu et al. [1] took. The system was trained and tested on both Dinu’s and PanLex test sets, and in both cases the matrices trained on Dinu’s data were the ones reaching higher scores. On the PanLex data experiments with different training set sizes were executed, out of which the 3K training set gave the best results. Continuing the training of the matrices obtained by using Dinu’s data with the PanLex dataset brought a slight improvement on the Italian-English scores, but English-Italian scores only got worse.

Finally, the system was trained on three different languages at the same time. The obtained pairwise precision values are proved to be worse than the results obtained when the system was trained in bilingual mode. However, these results are still promising considering that a completely new approach was implemented, and they showed that the system definitely learned from a data which is available for a wide range of languages.

The approach is quite promising but in order to reach state-of-the-art performance the system has to deal with some mathematical issues, for example dimension reduction in the universal space. Further experimentation in multilingual mode with an extended number of languages could also provide meaningful outputs. By involving expert linguistic knowledge various sets of languages could be constructed using either only very close languages, or, on the contrary, using very distant languages.
Thanks to the PanLex database, bilingual dictionaries can easily be extracted, which can, then, be directly used for multilingual experiments.

6. Acknowledgements

This work is a collaboration of the Universitat Politècnica de València (UPV) and the Budapest University of Technology and Economics (BUTE). Work partially supported by the Spanish MINECO and FEDER founds under project TIN2017-85854-C4-2-R.

7. References

TransDic, a public domain tool for the generation of phonetic dictionaries in standard and dialectal Spanish and Catalan

Juan-María Garrido¹, Marta Codina², Kimber Fodge²

¹Phonetics Lab, National Distance Education University, Spain
²Department of Translation and Language Sciences, Pompeu Fabra University, Spain
jmgarrito@flog.uned.es

Abstract

This paper presents TransDic, a free distribution tool for the phonetic transcription of word lists in Spanish and Catalan which allows the generation of phonetic transcription variants, a feature that can be useful for some technological applications, such as speech recognition. It allows the transcription in both standard Spanish and Catalan, but also in several dialects of these two languages spoken in Spain. Its general structure, input, output and main functionalities are presented, and the procedure followed to define and implement the transcription rules in the tool is described. Finally, the results of an evaluation carried for both languages are presented, which show that TransDic performs correctly the transcription tasks that it was developed for.

Index Terms: Automatic phonetic transcription, Speech recognition, Dialects, Spanish, Catalan, Tools

1. Introduction

Phonetised dictionaries are language resources that are useful for several purposes in Speech Technologies, but probably its most frequent use is in automatic speech recognition. Phonetised dictionaries included in speech recognition systems are expected to contain not a unique transcription for each entry in the dictionary, but several transcription variants representing alternative pronunciations of the word in the language, as it has been shown that including these variants in the dictionary improves speech recognition accuracy (see for example [1], among many other works). These variants may represent either dialectal, sociolectal or even individual variations of the pronunciation of the word.

Two main approaches to the generation of these dictionaries have been attempted: data-driven and knowledge-based [2]. Data-driven techniques obtain the transcription variants, as well as their relative frequency, from the analysis of a phonetised corpus, which has to be large and representative enough to derive a reliable set of pronunciation variants. This analysis has been carried out frequently using machine learning techniques ([3,4,5,6,7], among others). Despite its evident advantages, the main drawback of this approach is that it is very dependent on the contents of the used corpus, in which probably rare words in the language won’t be represented. Knowledge-based approaches make use of explicit linguistic knowledge, for example, in the form of transcription rules implemented in an automatic phonetic transcriber. In this case, the automatic transcriber generates one or several transcription variants for each word, depending on the number of phonetic processes that can be applied to it. The main advantage of this approach is that it ensures a full coverage of the considered pronunciation variants, but it has also disadvantages, such as the fact that an excessive number of variants may be generated for some words, which may cause a decreasing of the performance of the speech recognizer ([8]). Also, the probability of each transcription variant should be also obtained by rule, a much more difficult task in the current state of linguistic knowledge.

Other possible technological applications of phonetised dictionaries, such as the phonetisation of classical language dictionaries, do not require transcription variants. In this case, only a single transcription, corresponding to the standard pronunciation, is needed.

Most existing phonetic transcribers for Spanish and Catalan have been developed for a specific system of application, usually commercial, and they are not public domain ([9], for example). Most non-commercial transcribers available for these two languages ([10,11]) have limitations to its use (for example, they do not allow the transcription of items that are not words, they only handle a phonetic transcription alphabet—IPA or SAMPA—or they only allow the transcription through a web interface) which makes difficult the task of phonetising word lists. Also, they usually do not allow the transcription in other dialects different to the standard one. Saga [12], for Spanish, and Segre [13] for Catalan, do allow the generation of both standard and dialectal phonetic transcriptions, but they do not allow the simultaneous generation of alternative transcriptions for a single item. TransText [14], a phonetic transcriber for Spanish and Catalan, has the same limitation.

This paper presents TransDic, a free distribution tool for the phonetic transcription of word lists in both standard Spanish and Catalan, but also in several dialects spoken in Spain of these two languages. Its main novel feature is that it allows the generation of dictionaries both with one single transcription per entry or with several phonetic transcription variants, which makes it suitable for the generation of dictionaries for speech recognition systems. Also, it has been designed to generate only ‘relevant’ dialectal variants, that is, those which are general enough in the corresponding geographical area. It has been developed from TexAFon [15,16], a complete rule-based linguistic processing system for text-to-speech that includes a phonetic transcriber, and which has been improved to include new features, such as the generation of phonetic variants for a single input item and the transcription in several non-standard dialects.

Next sections present the general structure of TransDic and its main functionalities, and explain the procedure followed to define and implement in the tool the transcription rules. The results of an evaluation carried for both languages are also presented.
2. General description

TransDic is a multiplatform tool that can be used either in Linux, Mac OS or Windows. It is a command-line tool, which requires to specify a set of arguments for its execution (for example, the language or dialect or the phonetic alphabet for the transcription). It can be used then both as an independent tool or integrated in other tools or procedures.

TransDic internal structure includes three levels, as illustrated in Figure 1:

- The tool itself, TransDic.
- The processing core, shared by other applications (such as TransText, for the phonetic transcription of texts, or texafon, a full text-processing tool for text-to-speech applications), which includes the letter-to-sound module, other text processing modules, some of them (the text processing module, for example) used also by TransDic.
- A set of language/dialect modules, including language or dialect dependent dictionaries and rules. Every dialect is considered then as an ‘autonomous’ language, with its own rules and dictionaries.

![Figure 1: TransDic structure.](image)

The transcription procedure in TransDic involves three main stages, which are similar to the ones in other existing transcription tools:

1) Text preprocessing
2) Lexical stress prediction
3) Phonetic transcription

Unlike other available phonetic transcription tools, TransDic is able also to transcribe non-standard tokens (such as symbols, dates, times or other figures), which are converted to readable Spanish items by its text processing module.

TransDic input must be a UTF-8 text file containing the list of tokens (one line per token) to be transcribed. Arguments allow to specify, for example, the dialect in which transcription has to be produced:

- Catalan: Standard (ca), Ribagorza (ca-r), Pallars (ca_pa), Tortosa (ca_to), Central Western (ca_ac), Northern Valencian (ca_vs), Central Valencian (ca_vc), Southern Valencian (ca_vm) and Alicante Valencian (ca_al)
- Spanish: Peninsular Standard (es), Western Andalucía (es_aoc), Eastern Andalucía (es_aor), Extremadura North (es_exn), Extremadura South (es_exs), Canarias (es_can), Castilla-La Mancha (es_clm), Madrid (es_mad) and Murcia (es_mur)

Other arguments allow also to specify the phonetic alphabet to be used (IPA or SAMPA), if the transcription may include or not pronunciation variants and syllable marks or the format of the output dictionary.

It is important to note that the definition of the pronunciation variants to be considered in transcription is not specified directly using arguments, as in Saga, for example. In this case the selection of a given dialect determines the phonetic phenomena and pronunciation variants that will be taken into account.

The output of TransDic is another UTF-8 text file containing the phonetised dictionary, which can be generated in two different formats: default (Figure 2) or HTK (Figure 3).

![Figure 2: Sample of default output dictionary generated by TransDic.](image)

3. Development of the dialect transcription modules

Earlier versions of TexAFOm [15, 16] included only language modules for standard Catalan and Spanish. The development of TransDic has been possible after the development of a set of new language modules for the TexAFOm processing core, covering several Spanish and Catalan dialects spoken in Spain. Each language/dialect module includes a pronunciation exception dictionary and the transcription rules, written in Python, for the corresponding dialect, and they have been developed from the corresponding standard version. The detailed description of the development procedure of these rules exceeds the scope of this paper, but it can be summarised in the following steps for each dialect:

1. Definition of the phonetic phenomena characterising its pronunciation
2. Implementation of the phonetic rules covering the selected phenomena
3. Evaluation of the transcription and fixing of errors

These three steps are described briefly in the following subsections. A detailed description of the development procedure for the different dialect modules in Spanish and Catalan can be found in [17] and [18], respectively.
3.1. Phonetic characterization of dialects

Existing literature on dialectological studies of Spanish spoken in Spain ([19,20,21,22], among many others) and Catalan ([23], a recent work also among many others) usually describes dialectal pronunciations in a very detailed way, paying attention to both general and local phenomena. For this work, however, only those phenomena which are general enough in the geographical area of the dialect were considered. And within these general phenomena, a distinction was made between ‘primary’ (the most frequent in their corresponding geographical area) and ‘secondary’ (not as frequent as primary ones, but general enough to be taken into account in a general description of the dialect in question), in order to keep the number of pronunciation variants within a reasonable number. Only geographical variants were considered, not social, stylistic nor individual.

The goal of this phase was then to define a set of ‘primary’ and ‘secondary’ pronunciations for each considered dialect, and to establish the relations between them (that is, secondary pronunciations are alternative pronunciations to the primary ones). This task was carried out through an extensive literature review for both languages. It was not an easy task at all from a linguistic point of view, as the information provided in the literature is frequently incomplete, with limited information about the frequency or extension of the described phenomena.

The detailed description of all the defined dialect sets is out of the scope of this paper, so only one is presented here in some detail, the one for Canarias Spanish, based in the description provided mainly in [19,21,24,25]. Table 1 presents the subset of primary pronunciations which are different from standard Spanish in this dialect, the subset of secondary pronunciations and the relation between both subsets (that is, which secondary realisations are pronunciation variants of the primary ones). As it can be observed in this table, the relation between primary and secondary pronunciations can follow different patterns:

- One primary pronunciation has no secondary pronunciation associated, which means that no alternative realisations are considered. This is the case, for example, of the *seseo*.

- One primary pronunciation has one (or more) secondary pronunciation(s) associated. This means that all pronunciations, primary and secondary ones, are possible in the same context, although the primary one is considered as more frequent than the secondary one(s). For example, elision of syllable-final orthographical *<s>* (secondary pronunciation) is a possible alternative realisation to the pronunciation as [h], considered primary (that is, most frequent) in Canarias Spanish.

- One secondary pronunciation has no primary pronunciation associated. This means that it is an alternative to a standard pronunciation, which is also a primary pronunciation in that dialect. This is the case of *yeísmo*, the realisation of orthographical *<ll>* as [ʎ], which according to the literature has been considered as less frequent than the realisation as [l], default pronunciation in standard Spanish.

These lists of phonetic phenomena were used for the implementation phase, explained in the following subsection.

<table>
<thead>
<tr>
<th>Primary</th>
<th>Secondary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pronunciation of orthographical</td>
<td>Elision of syllable-final</td>
</tr>
<tr>
<td><em>&lt;c, z&gt;</em> as [s] (<em>seseo</em>)</td>
<td>orthographical <em>&lt;s&gt;</em></td>
</tr>
<tr>
<td><em>&lt;casa&gt;</em> → [kasa]</td>
<td><em>&lt;los&gt;</em> → [lo]</td>
</tr>
<tr>
<td>Pronunciation of orthographical</td>
<td>Pronunciation of interlocu-</td>
</tr>
<tr>
<td><em>&lt;s&gt;</em> at the end of a syllable</td>
<td>tive <em>&lt;d&gt;</em> in words</td>
</tr>
<tr>
<td>as [h] (<em>aspiración</em>)</td>
<td>ending with <em>&lt;ado&gt;</em></td>
</tr>
<tr>
<td><em>&lt;ajo&gt;</em> → [‘aho]*</td>
<td><em>&lt;cansado&gt;</em> → [kanˈsaðo]</td>
</tr>
<tr>
<td>Elision (no pronunciation)</td>
<td>Elision of interlocutive</td>
</tr>
<tr>
<td>orthographical <em>&lt;d&gt;</em> in words</td>
<td><em>&lt;d&gt;</em> as [d], as in stand-</td>
</tr>
<tr>
<td>ending with <em>&lt;ado&gt;</em></td>
<td>ard Spanish</td>
</tr>
<tr>
<td><em>&lt;cansado&gt;</em> → [kanˈsaðo]</td>
<td><em>&lt;comido&gt;</em> → [koˈme]</td>
</tr>
<tr>
<td>Elision of word-final <em>&lt;r,l&gt;</em></td>
<td>Pronunciation of word-final</td>
</tr>
<tr>
<td><em>&lt;comer&gt;</em> → [koˈre]</td>
<td><em>&lt;r,l&gt;</em> as [r,l], as in</td>
</tr>
<tr>
<td><em>&lt;Raquel&gt;</em> → [raˈke]</td>
<td>standard Spanish</td>
</tr>
<tr>
<td><em>&lt;cansado&gt;</em> → [kanˈsaðo]</td>
<td><em>&lt;comido&gt;</em> → [koˈme]</td>
</tr>
<tr>
<td>Elision of word-final <em>&lt;d&gt;</em></td>
<td>Pronunciation of word-final</td>
</tr>
<tr>
<td><em>&lt;corredor&gt;</em> → [koˈre]</td>
<td><em>&lt;d&gt;</em> as [d], as in stand-</td>
</tr>
<tr>
<td></td>
<td>ard Spanish</td>
</tr>
<tr>
<td><em>&lt;corredor&gt;</em> → [koˈre]</td>
<td><em>&lt;corredor&gt;</em> → [koˈre]</td>
</tr>
<tr>
<td>Pronunciation of orthographical</td>
<td>Pronunciation of ortho-</td>
</tr>
<tr>
<td><em>&lt;ll&gt;</em> as [j] (<em>yeísmo</em>)</td>
<td>graphical <em>&lt;ll&gt;</em> as [l]</td>
</tr>
<tr>
<td><em>&lt;calle&gt;</em> → [‘kale]</td>
<td><em>&lt;calle&gt;</em> → [kaˈle]</td>
</tr>
</tbody>
</table>

3.2. Implementation

The implementation of the new dialect modules was done in all cases taking as starting point the rules and dictionaries for the corresponding standard dialect and then making the necessary modifications to include the defined primary and secondary phenomena. Some changes in the language-independent core of TexAfon were done also to allow the generation of several pronunciation variants for a single input token.

To implement primary phenomena, new context-dependent Python rules were developed to replace the standard ones in those cases in which the primary pronunciation was different from the standard. Figure 4 presents an example of rule for a primary pronunciation in Andalusian Spanish. In some cases, the inclusion of these rules led to modify also the exception dictionary, to make some entries coherent with the new rule.

The implementation of the secondary phenomena was done in a second phase and involved the modification of the primary context-dependent rules to allow the generation of secondary transcriptions for the same context. Figure 5 illustrates the result of the modification of the same rule presented in figure 4 to include deletion of *<s>* as secondary variant.

Finally, if the user has specified with the corresponding argument that transcription variants should be generated, TransDiC produces all transcription variants for the input word. These transcription variants are derived from the character-by-character pronunciation variants generated by the transcription rules: the language-independent letter-to-sound...
module takes the obtained pronunciations for each character and combines them to generate the word transcription variants. So, for example, for the word ‘casas’ two different transcription variants would be generated ([kəsə], [kəsə]) using the previously described rules, whereas in the case of ‘llover’ the output variants would be four ([ʎəβe], [joβe], [ʎoβe], [joβe]), as the input word includes two characters with transcription variants which are combined to create the different alternative word transcriptions.

```python
if ch == "s" and nch == "NIL":
    salida.append(["hh",0,False])
return salida
```

![Figure 4: Python implementation of a phonetic transcription rule for the pronunciation of orthographical <s> as [h] (aspiración) at the end of a word in Andalucía Spanish (Fodge, 2014).](image)

```python
if ch == "s" and nch == "NIL":
    salida.append(["hh",0,False])
    # Update deletion of word final syllable final s
    salida.append(["",0,False])
return salida
```

![Figure 5: Python implementation of a secondary pronunciation transcription rule for <s> deletion (in bold) in Andalucía Spanish (Fodge, 2014).](image)

### 3.3. Evaluation

The procedure to evaluate the transcription produced by the new dialect modules was similar for Spanish and Catalan: lists of isolated words, representative of the implemented phenomena (307 words for Spanish; 99 for Catalan), were processed using each dialect module to obtain the corresponding output transcription. These output transcriptions were then revised manually to detect possible errors. Both evaluations were carried out using a different tool of the TexAFon package, but the evaluated rules were the same used in TransDic [17,18].

In the case of Spanish, the transcription performance was perfect: no errors were detected. Some errors were found, however, in the case of Catalan dialects. An error measure was computed in this case, using the procedure described in [26]: the sum of all phone substitutions (Sub), deletions (Del) and insertions (Ins) divided by the total number of phones in the reference transcription (N). Table 2 presents the obtained error values.

The number of generated variants was also evaluated. Phonetised dictionaries were generated with TransDic for all dialects using two reference word lists in Spanish (1,000 most frequent words in the CREA corpus [27]) and Catalan (a cleaned version of the CesCa corpus [28], 1,648 words) and the mean number of variants per entry was computed. Tables 3 and 4 present the results.

The results of these two evaluations indicate that TransDic performs reasonably well both as for transcription quality and number of generated variants is concerned. Spanish modules provide a more accurate transcription than Catalan ones, but they tend to generate more variants per input entry than Catalan modules. Anyway, mean number of variants per entry is always below 2 in both languages.

### Table 2: Error measures obtained for the Catalan dialects.

<table>
<thead>
<tr>
<th>Dialect</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ribagorza</td>
<td>8</td>
</tr>
<tr>
<td>Pallarés</td>
<td>11</td>
</tr>
<tr>
<td>Tortosa</td>
<td>9</td>
</tr>
<tr>
<td>Central area</td>
<td>8</td>
</tr>
<tr>
<td>North Valencia</td>
<td>9</td>
</tr>
<tr>
<td>Central Valencia</td>
<td>10</td>
</tr>
<tr>
<td>South Valencia</td>
<td>10</td>
</tr>
<tr>
<td>Alicante Valencian</td>
<td>9</td>
</tr>
</tbody>
</table>

### Table 3: Mean number of variants per entry obtained for the Spanish dialects.

<table>
<thead>
<tr>
<th>Dialect</th>
<th>Mean number of variants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>1.021</td>
</tr>
<tr>
<td>Western Andalucia</td>
<td>1.76</td>
</tr>
<tr>
<td>Eastern Andalucia</td>
<td>1.354</td>
</tr>
<tr>
<td>Extremadura North</td>
<td>1.354</td>
</tr>
<tr>
<td>Extremadura South</td>
<td>1.76</td>
</tr>
<tr>
<td>Canarias</td>
<td>1.468</td>
</tr>
<tr>
<td>Castilla-La Mancha</td>
<td>1.207</td>
</tr>
<tr>
<td>Madrid</td>
<td>1.207</td>
</tr>
<tr>
<td>Murcia</td>
<td>1.713</td>
</tr>
</tbody>
</table>

### Table 4: Mean number of variants per entry obtained for the Catalan dialects.

<table>
<thead>
<tr>
<th>Dialect</th>
<th>Mean number of variants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>1</td>
</tr>
<tr>
<td>Ribagorza</td>
<td>1.006</td>
</tr>
<tr>
<td>Pallarés</td>
<td>1</td>
</tr>
<tr>
<td>Tortosa</td>
<td>1</td>
</tr>
<tr>
<td>Central area</td>
<td>1</td>
</tr>
<tr>
<td>North Valencia</td>
<td>1.269</td>
</tr>
<tr>
<td>Central Valencia</td>
<td>1.003</td>
</tr>
<tr>
<td>South Valencia</td>
<td>1.003</td>
</tr>
<tr>
<td>Alicante Valencian</td>
<td>1.003</td>
</tr>
</tbody>
</table>

### 4. Conclusions

This paper has presented TransDic, a tool for the generation of phonetised dictionaries for Catalan and Spanish. Its most innovative features are that it allows to transcribe in several Spanish dialects spoken in Spain (Saga [12], for example, was developed considering mainly the American Spanish dialects) and that it allows the creation of phonetic dictionaries containing a reasonable number of pronunciation variants. The knowledge-based approach used in TransDic, based on a careful selection of the phonetic phenomena considered for transcription, does not lead to an overgeneration of variants, a classical problem in this kind of approach. Finally, another interesting feature of TransDic is that it is available for free download, from https://sites.google.com/site/juanmariagarriod/research/resources/tools/transdic. TransText [14], a phonetisation tool which allows the transcription of texts in the same dialects as TransDic, is also available for download from https://sites.google.com/site/juanmariagarriod/research/resources/tools/transnertext.

Some expected improvements for the tool in the future are the inclusion of new types of rules for variants (for example, rules for informal pronunciations), and a deeper evaluation of the output by native speakers of each dialect.
5. References


Wide Residual Networks 1D for Automatic Text Punctuation

Jorge Llombart, Antonio Miguel, Alfonso Ortega, Eduardo Lleida

ViVoLAB, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain
jlombg@unizar.es, amiguel@unizar.es, ortega@unizar.es, lleida@unizar.es

Abstract

Documentation and analysis of multimedia resources usually requires a large pipeline with many stages. It is common to obtain texts without punctuation at some point, although later steps might need some accurate punctuation, like the ones related to natural language processing. This paper is focused on the task of recovering pause punctuation from a text without prosodic or acoustic information. We propose the use of Wide Residual Networks to predict which words should have a comma or stop from a text with removed punctuation. Wide Residual Networks are a well-known technique in image processing, but they are not commonly used in other areas as speech or natural language processing. We propose the use of Wide residual networks because they show great stability and the ability to work with long and short contextual dependencies in deep structures. Unlike for image processing, we will use 1-Dimensional convolutions because in text processing we only focus on the temporal dimension. Moreover, this architecture allows us to work with past and future context. This paper compares this architecture with Long-Short Term Memory cells which are used in this task and also combine the two architectures to get better results than each of them separately.

Index Terms: Text Punctuation, Wide Residual Network, Recurrent Neural Networks, Natural Language Processing

1. Introduction

Nowadays, there are a great amount of multimedia resources from which we want to extract more and information. This usually requires a large pipeline with many stages where, at some point, it is needed to have accurate punctuation to accomplish later steps like machine translation, summarization, sentiment analysis, etc. This task is usually performed after some automatic speech recognition system, and it usually takes some prosodic information like pauses times. Some systems are based on a capitalization and punctuation system [1], on Long-Short Term Memory cells (LSTM) [2] or on Bidirectional Long-Short Term Memory cells (BLSTM) [3], but not always it is possible to access to these features. Other works do not use prosodic features. Some uses Conditional Random Fields (CRF) [4], or in a combination with LSTM [5]. There are approaches with Convolutional Neural Networks (CNN) [6], or distilling from an ensemble of different neural networks [7]. There are also works focused on Character-level Neural Networks [8], where they also use architectures with Convolutional Highway, Residual networks and BLSTM.

In image classification, convolutional neural networks have achieved very good results. In those tasks convolutional networks have gain depth, even hundreds of layers. This was possible with Residual Networks [9]. They use very deep and thin structures, but they suppose a great amount of computation. Wide Residual networks (WRN) were designed to obtain better results with less depth and, therefore with less computation, increasing the width [10, 11].

Wide Residual networks are also used in automatic speech recognition obtaining good results [12, 13]. In those architectures, Wide Residual Networks use two-dimensional convolutions, considering the cepstral input as an image, but in this study we consider that in speech and natural language processing the important dimension is the temporal dimension, and correlation along other dimensions is lower, so we use convolutions 1D.

This paper is divided in four sections. The first is the introduction. In the second section we describe the punctuation task and the neural network models used to perform this task. In section three we explain how the experiments are done. And in the last section we present some conclusions and future work.

2. Text punctuation modelling

In this paper, we will focus on recovering punctuation from Spanish texts where punctuation is missing. Our experiments are designed to recover periods and commas, and we simulate the task by deleting them from a collection of texts. In Spanish there are some different ways to express the end of a sentence: closing exclamation, closing interrogation, full stop, period, etc. In order to simplify the classification we consider that all kind of end of sentence is marked as period. So the set of labels in our experiments consists on \( L = \{"Space", "Period", "Comma", "Pad" \} \) where we will label each input word of our experiments as:

- word without punctuation mark: "Space"
- word followed by any kind of period: "Period"
- word followed by comma: "Comma"
- padding: "PAD".

This experiment uses a fixed dictionary \( O = \{0, \ldots, D+1\} \), with \( D \) the vocabulary size plus two extra symbols. There are two special words, first one is \( o_0 = "PAD" \) which is reserved to fill input sequences when it is necessary, and the second is \( o_2 = "UNK" \), which is reserved for those words not included in \( O \), usually called out of vocabulary words.

Text punctuation is a classification task in which we want to assign a label from \( L \) to each element of an input sequence \( X = \{x_0, \ldots, x_t, \ldots, x_T\} \) with \( t \in T \) the maximum number of elements in the input sequence. Each \( x_t \) corresponds with one word \( o_2 \in O \). Our Neural Network provide as output a sequence of labels \( Y = \{y_0, \ldots, y_t, \ldots, y_T\} \), \( Y_t \in L \). In Figure 1 we represent an input vector \( X_{in} \) with the training label vector \( Y_{True} \) that corresponds with it.

![Figure 1: Input sequence and its tag correspondence for an input sequence of \( T = 9 \) when the text only has 8 words.](image-url)
2.1. Wide Residual Networks 1D

WRNs are usually used in image tasks where two spatial dimensions are the context to work. In automatic speech recognition we can use spectrograms as an image to adopt this two-dimensional schema [12, 13]. However, in text processing, we work only with temporal dimension because word relations are only meaningful along this dimension. WRN is a structure based on 2D-convolutional networks. In order to adapt this structure to our environment, we will work with 1D-convolutional layers. In image context, convolutional layers are characterized by the number of output channels, \( C \), but in text context, we consider channels as a temporal sequence of vectors, where each element of this vectors is one of the \( C \) filters response for this sequence moment.

A WRN consists on several blocks, that in this paper we will call Wide Residual Block (WRB). In each of this WRB, we increase the number of channels in the output. We will widen the convolutional output. These WRBs are also constructed by several Residual Block (RB). This Blocks are the basic elements of the WRNs, and them have two paths. The first path consists on two convolutional layers with batch normalization [14] and ReLU non-linearity [15]. The second path is a residual connection between block input and output. In this work we use a sum of the block input and the convolutional path as residual connection. In the case where the number of channels of the two paths are different, we adjust the number of channels with a convolutional layer in the residual path. In Figure 2a we show the complete structure of the WRNs that we use in this work. We will characterize the WRB by its widen factor, which is the factor we apply to the number of channels at block input to gain width. In Figure 2a, we describe convolutional layers by its kernel dimension. This kernel dimension is the number of words that the convolutional filter takes to compute each element in the output. In this work the kernel dimension of the first convolution layer is \( 5 \). Convolutional layers in WRB use a kernel of dimension \( 3 \). There is a special case when we use a convolutional layer with kernel dimension of one. This kind of convolutional layer sometimes is called 1 x 1 convolution, position independent convolution or position wise fully connected. It is like a fully connected layer with all its weights shared through time. In other words, each element of the input is evaluated with same weights along all sequence. In this layers we also describe input dimension and output dimension of each sequence element as \( [\text{input dimension} \times \text{output dimension}] \).

Usually in classification we adopt the many-to-one structure where we use an input sequence to classify the central element of the sequence. In convolutional networks for classification, usually we use pooling and reduce operations in order to classify this central element. A problem that has this architecture to classify two contiguous elements, we need to process two sequences that only differ in one element. This produce that we recompute \( T-1 \) times the same element in the sequence. In order to reduce this massive excess of computation in long context sequences we will use a many-to-many paradigm. In our experiments we select unique very long sequences from the text as the context, and we compute the convolutions to the whole sequence only once. For this our WRNs do not use any pooling strategy, the stride is always 1 and use the appropriated padding in order to fix the edge effects in convolutional layers. Therefore, at the output of the WRBs \( X_{WRN} \) in figure 2a, we have the same sequence length as the input, \( T \). In Figure 2b it is represented the shape of the WRNs output sequence. \( N \) is the number of sequence examples evaluated at the same time in a forward pass, or also called mini-batch. Each sequence has \( T \) elements, where each element is a vector \( x_{t} \in \mathbb{R}_{n,t}, n \in \mathbb{N}, t \in \mathbb{T} \) with \( C_{W} \) the convolutional filter responses. In Figure 2b we show that each \( x_{n,t} \) is evaluated through the position independent convolution layer and a Softmax non-linearity to obtain the final classification.

To train the network we use the cross-entropy as cost function \( J(x, y) \). In the case of many-to-one structure we compute the mini-batch error as:

\[
J_{\text{many to one}} = \frac{1}{N} \sum_{n=0}^{N-1} J(x_{n}, y_{n}),
\]

where each example sequence of \( T \) elements correspond to one example of \( N \) in the mini-batch. In our case we implement the many-to-many structure. We also use the cross-entropy error loss function \( J(x, y) \), but the mini-batch cost is computed for each of the \( T \) elements of each of the \( N \) examples of the mini-batch as:

\[
J_{\text{many to many}} = \frac{1}{N} \sum_{n=0}^{N-1} \frac{1}{T} \sum_{t=0}^{T-1} J(x_{n,t}, y_{n,t})
\]

2.2. Long-Short Term Memory Cells

In this study we will compare WRN with LSTM structures and BLSTM structures. In these cases we use the same input \( X_{n} \) as described in section 2. In Figure 3 we show that the the input sequences are processed by an LSTM or BLSTM obtaining a sequence with the same length as the input. In the case of a LSTM or a BLSTM, the sequence \( X_{n} \) passes through the embedding layer as in a WRN obtaining a sequence of same length but each element \( x_{t} \) from the sequence is a vector of embedding dimension \( C \), as we represent in Figure 3. This sequence is evaluated by the LSTM or the BLSTM obtaining a sequence where each element has dimension \( H \) or \( 2 \times H \) respectively with \( H \) the hidden dimension of the LSTM or the BLSTM layers. With this structure we use the same philosophy many-to-many as in WRN, and we use the same loss function described in equation 2.

To obtain a combination of WRN and BLSTM we just substitute the position independent convolution and the Softmax in WRN by the BLSTM structure showed in Figure 3. Then we will remove or add WRBs to the structure to get different results discussed in section 3

3. Experiment and Results

For this experiment we have collected 1,181,413 articles from electronic edition of 32 diverse Spanish magazines and newspapers, form January 2017 up to February 2018. Each article has 506 words on average. For our purposes all texts were case lowered; numbers, dates, hours and Roman numbers were automatically transcribed and normalized; some units and abbreviations like Km, m, Kg, min, were changed by their transcription; all symbols were deleted; and final question and exclamation marks were substituted by dots. This data set was segmented in train set (980,000 articles), development set (101,000 articles) and test set (100,413 articles), randomly chosen. The dictionary for this work is composed by all words in train text which appear at least 50 times in order to avoid misspelled and rare words. This supposes that all our experiments use a dictionary with 116,737 words including especial delimiters.

In all of the architectures tested in this work we use as first layer an embedding layer with input dimension the number of
In this paper we start considering as baseline an LSTM and a BLSTM. In Table 1 we show that BLSTM provides better results.

In order to understand the behaviour of each architecture, we have studied the response of the output when we modify the input. We want to get an idea of the contribution of each word from the input to the classification of a particular output time. For this we have cancelled sequentially each word of the input, by forcing to zero the output of the embedding layer for this word, and representing the value of the network output after a forward pass of this modified input. We assume that those words that do more contribution to a correct classification would be those words that when are cancelled disturb more the classification. In Figure 4 we can see the output values for "Period" class before the Softmax non-linearity of the LSTM, BLSTM and WRN for an output time where a "Period" should be predicted. The dot probability output is shown as each word of the input is sequentially cancelled. One of the first things we can look at is the temporal distance of the first and last cancelled word that does a significant perturbation in "Period" classification. LSTM only has perturbations with words before the moment in which we want the classification. We can see that this architecture only uses past information for classification. This behavior may lead to worse performance because future words should be useful in punctuation task. We can not say that a sentence is ended if we do not know when the text change of sense, or subject, or action. In BLSTM architecture we can see that it uses words from past and from future. Even those words that do a major perturbation are from a future context. The case of WRNs is the same as BLSTM, but it uses less number of words for the classification because the perturbations are closer to the classification time.

For evaluation purposes we use three measures, precision, recall and $F_1$-measure, for each class of interest.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (3)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (4)$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (5)$$

For this measures only relevant cases are taken into account as we see in equations 3, 4 and 5, where only consider the True Positive, False Positive, and False Negative for each class of interest.

For this paper we start considering as baseline an LSTM and a BLSTM. In Table 1 we show that BLSTM provides better results.
than LSTM achieving a 88.03 of $F_1$ with “Period” class, and a 90.53 of $F_1$ with “Comma”. These improvements are obtained thanks to use past and future context in BLSTM.

Residual connections usually work better when the network is very deep, but they also help in shallower networks. In order to see how residual connections help the classification, we trained convolutional networks which are exactly the same architecture as WRNs but without residual connections. In Table 1 we show that when we increase the number of blocks in convolutional networks without residual connections, they start to get worse results than the same configuration in WRNs. With only one block they perform equal, but with three blocks, the network without residual connections performs worse. We achieve the best result with a WRN using three blocks. We obtain an $F_1$ of 88.21 for “Period” and 90.50 for “Comma”, which is the same performance as the base case of a BLSTM.

In the last experiments we concatenate the output of a WRN with the input, without embedding layer, of the BLSTM. These two architectures are compatibles because both are designed to work with sequences, processing element by element, so we do not need to do any transformation to the inputs or outputs. In Table 1 we present the result of concatenation of a WRN of one, two or three blocks and a BLSTM with the same architecture as the baseline. We can see that all those configurations perform better than any of the two architectures alone. The case of a WRN of one block presents the better result with an $F_1$ of 90.82 in “Period” and 92.60 in “Comma”.

### 4. Conclusions and future work

In this work, we have shown the use of Wide Residual Networks for punctuation recovering. We show that architectures that use past and future context obtain better results. WRNs perform similarly to BLSTM in our experiments so other aspects of the architecture should be taken into account, like training time or resources consumed. The best results are obtained when WRNs and BLSTMs are concatenated. This work suggests the idea of the use of WRNs as feature extractor. Then we need powerful structures than a linear classifier to perform the final classification.

In problems that require sequential processing, BLSTM structures that use past and future context, are the best option.

For future work, it would be interesting to differentiate between question marks and “periods”. In Spanish, this is a challenging task since we need to predict the closing question mark, but also the opening question mark, which is a symbol not used in other languages like English. Future work also will pay attention to architecture improvements like hyper-parameter configuration or to include attention mechanisms.

### 5. Acknowledgements

This work is supported by the Spanish Government and the European Union under project TIN2017-85854-C4-1-R, and by Gobierno de Aragón / FEDER (research group T36_17R). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.
6. References


End-to-End Multi-Level Dialog Act Recognition

Eugénio Ribeiro¹², Ricardo Ribeiro¹³, and David Martins de Matos¹²

¹²F – Spoken Language Systems Laboratory – INESC-ID Lisboa, Portugal
²Instituto Superior Técnico, Universidade de Lisboa, Portugal
³Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

eugenio.ribeiro@12f.inesc-id.pt

Abstract

The three-level dialog act annotation scheme of the DIHANA corpus poses a multi-level classification problem in which the bottom levels allow multiple or no labels for a single segment. We approach automatic dialog act recognition on the three levels using an end-to-end approach, in order to implicitly capture relations between them. Our deep neural network classifier uses a combination of word- and character-based segment representation approaches, together with a summary of the dialog history and information concerning speaker changes. We show that it is important to specialize the generic segment representation in order to capture the most relevant information for each level. On the other hand, the summary of the dialog history should combine information from the three levels to capture dependencies between them. Furthermore, the labels generated for each level help in the prediction of those of the lower levels. Overall, we achieve results which surpass those of our previous approach using the hierarchical combination of three independent per-level classifiers. Furthermore, the results even surpass the results achieved on the simplified version of the problem approached by previous studies, which neglected the multi-label nature of the bottom levels and only considered the label combinations present in the corpus.

Index Terms: dialog act recognition, DIHANA corpus, multi-level classification, multi-label classification

1. Introduction

Dialog act recognition is an important task in the context of Natural Language Understanding (NLU) since dialog acts reveal the intention behind the uttered words, allowing the application of specialized interpretation approaches. Consequently, it has been widely explored over the years on multiple corpora with different characteristics [1]. In this sense, the distinguishing aspect of the DIHANA corpus [2], which features interactions in Spanish between humans and a train information dialog system, is its three-level annotation scheme. While the top level refers to the generic task-independent dialog act, the others complement it with task-specific information. Additionally, while each segment has a single top-level label, it may have multiple or no labels on the other levels. Thus, the DIHANA corpus poses both multi-level and multi-label classification problems. However, most previous studies on this corpus approached the task as a single-label classification problem in which the label of a segment was the combination of all its labels. Contrarily, in [3] we explored each level independently, approaching the bottom levels as multi-label classification problems, and then combined the best classifiers for each level hierarchically. Among other conclusions, in that study we have shown that there are dependencies between the multiple levels which the independent classifiers cannot capture unless the information is explicitly provided. Thus, in this paper we approach the problem using an end-to-end classifier to predict the labels of the three levels in parallel, so that the relations between the levels are captured implicitly. Additionally, we explore approaches on segment and context information representation which have recently been proved successful on the dialog act recognition task and were not used in our previous study on the DIHANA corpus.

In the remainder of the paper we start by providing an overview of previous work on dialog act recognition on the DIHANA corpus and how it can be improved, in Section 2. Then, in Section 3, we describe our experimental setup, including the corpus, the network variations, and the training and evaluation approaches. The results of our experiments are presented and discussed in Section 4. Finally, Section 5 states the most important conclusions of this study.

2. Related Work

Automatic dialog act recognition is a task that has been widely explored using multiple classical machine learning approaches, from Hidden Markov Models (HMMs) to Support Vector Machines (SVMs) [1]. However, recently, most approaches on the task take advantage of Deep Neural Network (DNN) architectures to capture different aspects of the dialog [4, 5, 6, 7, 8, 9].

Similarly to the studies on English data, the first studies on the DIHANA corpus employed HMMs using both prosodic [10] and textual [11] features. The study using prosodic features focused on the prediction of the generic top level labels, while the study using textual features considered the combination of the multiple levels. Additionally, the latter study, as well as a more recent one [12], explored the recognition of dialog acts on unsegmented turns using n-gram transducers. However, in those cases, the focus was on the segmentation process. The results of the HMM-based approaches were surpassed in a study that applied SVMs [13] to a feature set consisting of word n-grams, the presence of wh-words and punctuation, and context information from up to three preceding segments in the form of the same features. All of these studies neglected the multi-label nature of the bottom levels of the dialog act annotation scheme of the DIHANA corpus and approached a simplified single-label problem in which the label set consisted of the label combinations present in the corpus. However, this approach limits the possible combinations to those existing in the dataset and neglects the distinguishing characteristics of each individual label.

Contrarily to those studies, in [3] we explored each level independently, approaching the bottom levels as multi-label classification problems, and then combined the best classifiers for each level hierarchically. In that study we compared those that were the two top performing DNN-based approaches on segment representation for dialog act recognition on the Switchboard Dialog Act Corpus [14], which is the most explored cor-
pus for the task. One of those approaches uses a stack of Long Short Term Memory (LSTM) units to capture long distance relations between tokens [7], while the other uses multiple parallel Convolutional Neural Networks (CNNs) with different context window sizes to capture different functional patterns [8]. We have shown that the CNN-based approach leads to better results on every level of the DIHANA corpus. However, while wider context windows are better for predicting the generic dialogue acts of the top level, the task-specific bottom levels are more accurately predicted when using narrower windows. Furthermore, recently, we have shown that the performance can be improved by using a Recurrent Convolutional Neural Network (RCNN)-based segment representation approach that is able to capture long distance relations and discards the need for selecting specific window sizes for convolution [9]. Additionally, we have shown that a character-based segment representation approach achieves similar or better results than an equivalent word-based approach on the Switchboard corpus and the top level of the DIHANA corpus and that the information captured by both approaches is complementary [15].

In [3] we have also shown that context information concerning the dialog history and the classification of the upper levels is relevant for the task. Concerning the dialog history, we have explored the use of information from up to three preceding segments in the form of their classifications. On the first two levels, similarly to what happened in previous studies on the influence of context on dialog act recognition [16, 8], we have observed that the first preceding segment is the most important and that the influence decays with distance. On the other hand, since the bottom level refers to information that is explicitly referred to in the segment, it is not influenced by information from the preceding segments, at least at the same level. Recently, we have shown that the representation of information from the preceding segments used in previous studies does not take the sequentiality of those segments into account and that the whole dialog history can be summarized in order to capture that information as well as relations with more distant segments [9]. Additionally, in this paper, we further explore the relations between levels by using an end-to-end approach to predict the labels of the three levels in parallel and capture those relations implicitly.

3. Experimental Setup

This section presents our experimental setup, starting with a description of the corpus, followed by an overview of the aspects addressed by our experiments and the used network architecture and a description of the training and evaluation approaches.

3.1. Dataset

The DIHANA corpus [2] consists of 900 dialogs between 225 human speakers and a Wizard of Oz telephonic train information system. There are 6,280 user turns and 9,133 system turns, with a vocabulary size of 823 words. The turns were manually transcribed, segmented, and annotated with dialog acts [17]. The total number of annotated segments is 23,547, with 9,715 corresponding to user segments and 13,832 to system segments.

The dialog act annotations are hierarchically decomposed in three levels [18]. The top level, Level 1, represents the generic intention of the segment, while the others refer to task-specific information. There are 11 Level 1 labels, out of which two are exclusive to user segments and four to system segments. Overall, the most common label is Question, covering 27% of the segments, followed by the Answer and Confirmation labels, covering 18% and 15%, respectively. This is consistent with the information-transfer nature of the dialogs.

Although they share most labels, the two task-specific levels focus on different information. While Level 2 is related to the information that is implicitly focused in the segment, Level 3 is related to the kind of information that is explicitly referred to in the segment. There are 10 labels common to both levels and three additional ones on Level 3. The most common Level 2 labels are Departure Time, Fare, and Day, which are present in 32%, 14%, and 8% of the segments, respectively. On the other hand, the Level 3 label distribution is more balanced, with the most common labels, Destination, Day, and Origin, being present in 16%, 16%, and 13% of the segments, respectively.

While a segment has a single Level 1 label, it may have multiple or no labels in the other levels. In this sense, only 63% of the segments have Level 2 labels, and that percentage is even lower, 52%, when considering Level 3 labels. This is mainly due to the fact that Level 1 labels concern dialog structuring or communication problems cannot be paired with any labels in the remaining levels.

3.2. End-to-End Neural Network Architecture

In our experiments, we incrementally built the architecture of our network by assessing the performance of different approaches for each step. However, due to space constraints, we are not able to show individual figures for all of those approaches. Thus, in Figure 1 we show the architecture of the final network and use it to refer to the alternatives we explored.

At the top are our two complementary segment representation approaches. On the left is the word-based approach, which captures information concerning both word sequences and functional patterns using the adaptation of the RCNN by Lai et al. [19] which we introduced in [9]. In our adaptation we replaced the simple Recurrent Neural Networks (RNNs) used to capture the context surrounding each token by Gated Recurrent Units (GRUs), in order to capture relations with more distant tokens. To represent each word, we use a 500-dimensional Word2Vec [20] embeddings trained on the Spanish Billon Word Corpus [21]. On the right is the character-based approach we introduced in [15], which uses three parallel CNNs with different window sizes to capture relevant patterns concerning affixes, lemmas, and inter-word relations. We performed experiments using each approach individually, as well as their combination, which is depicted in Figure 1.

The representation of the segment can then be combined with context information concerning the dialog history and speaker changes. We provide the latter in the form of a flag stating whether the speaker changed in relation to the previous segment, as in [8, 9]. To provide information from the preceding segments, we use the approach we introduced in [9], which summarizes the dialog history by passing the sequence of dialog act labels through a GRU. We performed experiments using a single summary combining information concerning the three levels, as well as using per-level summaries which summarize the sequence of preceding labels of each level individually.

To predict the dialog act labels for the segment, the combined representation is passed through two dense layers. While the first reduces its dimensionality and identifies the most relevant information present in that representation, the second generates the labels. In terms of the dimensionality reduction layer, we experimented using a single layer that captures the most relevant information that is generic to the three levels, as well as per-level dimensionality reduction layers, which capture the
most relevant information for each level, as depicted in Figure 1. Since the first level poses a single-label classification problem, the output layer uses the softmax activation and the categorical cross entropy loss function. On the other hand, since the other levels pose multi-label classification problems, the corresponding output layers use the sigmoid activation and the binary cross entropy loss function which, given the possibility of multiple labels, is actually the Hamming loss function [22]. In both cases, for performance reasons, we use the Adam optimizer [23].

In [3], we have shown that the prediction of dialog act labels of a certain level is improved when information concerning the upper levels is available. Thus, as shown in Figure 1, we also performed experiments that considered the output from the upper levels in the dimensionality reduction layers.

3.3. Training and Evaluation
To implement our networks we used Keras [24] with the TensorFlow [25] backend. We used mini-batching with batches of size 512 and the training phase stopped after 10 epochs without improvement. The results presented in the next section refer to the average ($\mu$) and standard deviation ($\sigma$) of the results obtained over 10 runs. On each run, we performed 5-fold cross-validation using the folds defined in the first experiments on the DIHANA corpus [10, 11]. In terms of evaluation metrics we use accuracy. This metric is penalizing for the multi-label classification scenarios of Level 2 and Level 3, since it does not account for partial correctness [26]. However, due to space constraints and since we are focusing on the combined prediction of the three levels, we do not report results for specialized metrics.

4. Results
In this section we present the results of our experiments on each level, as well as on the combination of the three levels. Both the results on Level 1 and on the combination of the three levels can be directly compared with those reported in [3]. However, that is not the case for the remaining levels. In [3], since we explored each level independently and the annotation scheme does not allow segments with a Level 1 label concerning dialog structuring or communication problems to have labels in the remaining levels, we did not consider those segments when training and evaluating the Level 2 and Level 3 classifiers. Contrarily, since in this study we use a single classifier to predict the labels for all levels, those segments are also considered.

We started by exploring the word- and character-based segment representation approaches, as well as their combination. Thus, in these experiments, we did not provide context information to the network and we used a single dimensionality reduction layer for the three levels. In Table 1 we can see that, as we have previously shown in [15], the character-based approach leads to better results than the word-based one on Level 1. Additionally, the results achieved by the word-based approach are above those reported in [15], which confirms that the word-level RCNN-based segment representation approach leads to better results than the CNN-based one we used in [15] and [3]. On the other hand, the slight performance decrease of the character-based approach can be explained by the combined prediction of the three levels, which does not allow the classifier to specialize in predicting Level 1 labels. On the remaining levels, the character-level approach still performs better. However, the difference is smaller than on Level 1, which is explained by the more prominent relation of the labels of these levels with specific words. Furthermore, the combination of both approaches leads to the best results on every level.

Table 1: Accuracy (%) results according to the segment representation approach.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Level 1 $\mu$</th>
<th>Level 1 $\sigma$</th>
<th>Level 2 $\mu$</th>
<th>Level 2 $\sigma$</th>
<th>Level 3 $\mu$</th>
<th>Level 3 $\sigma$</th>
<th>All $\mu$</th>
<th>All $\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word-Based</td>
<td>92.18</td>
<td>.13</td>
<td>79.36</td>
<td>.18</td>
<td>79.35</td>
<td>.20</td>
<td>75.82</td>
<td>.17</td>
</tr>
<tr>
<td>Character-Based</td>
<td>95.31</td>
<td>.07</td>
<td>81.25</td>
<td>.47</td>
<td>81.24</td>
<td>.54</td>
<td>78.67</td>
<td>.51</td>
</tr>
<tr>
<td>Combined</td>
<td>95.64</td>
<td>.07</td>
<td>82.46</td>
<td>.20</td>
<td>82.44</td>
<td>.21</td>
<td>79.88</td>
<td>.17</td>
</tr>
</tbody>
</table>

By using per-level dimensionality reduction layers, the classifier is able to select the information that is most relevant for predicting the labels of each level. Thus, as shown in Table 2, this adaptation leads to improved results on the two bottom levels and on the combination of the three levels. However, the performance on Level 1 did not improve, which suggests that the combined segment representation captures more information concerning specific words in detriment of functional patterns relevant for the prediction of some Level 1 labels. Providing information concerning the output generated for the upper levels leads to further improvement, in line with that reported in [3] in spite of not using gold standard labels.

As stated in Section 2, context information from the pre-
Table 2: Accuracy (%) results according to the dimensionality reduction approach.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>μ</td>
<td>σ</td>
<td>μ</td>
<td>σ</td>
</tr>
<tr>
<td>Single Reduction</td>
<td>95.64</td>
<td>.07</td>
<td>82.46</td>
<td>.20</td>
</tr>
<tr>
<td>Per-Level Reduction</td>
<td>95.64</td>
<td>.05</td>
<td>83.21</td>
<td>.11</td>
</tr>
<tr>
<td>Output Waterfall</td>
<td>95.65</td>
<td>.05</td>
<td>83.29</td>
<td>.21</td>
</tr>
</tbody>
</table>

As shown in previous studies, using information concerning speaker changes slightly improves the performance, up to 95.64% average accuracy on the combination of the three levels. More importantly, as discussed in [3], the system segments are scripted and, thus, are easier to predict than the user segments. Furthermore, a dialog system is aware of its own dialog acts and must only predict those of its conversational partners. As expected, the performance decreases if the classifier is trained and evaluated on user segments only. The average decrease on the combination of the three levels is of 4.5 percentage points. However, on Level 1 it is of just .67 percentage points.

Since we use a single classifier to predict the labels for the three-levels, there is no explicit restriction that segments with Level 1 labels concerning dialog structuring or communication problems cannot have labels in the remaining levels. However, if we post-process the results to enforce that restriction, the improvement on the combination of the three levels is of just .03 percentage points when considering all segments and .1 when considering user segments only. This shows that the network is able to learn that restriction based on the training examples.

Overall, the average accuracy of our best approach on the combination of the three-levels is 95.67%. This result is 3.33 percentage points above the 92.34% we achieved in [3] using the hierarchical combination of independent classifiers for each level. Furthermore, it is even above the 93.98% achieved when considering the single-label simplification of the problem, which only considers the label combinations present in the corpus. This shows that the network is able to capture relevant relations between levels while still being able to identify the most important information for each level using the per-level dimensionality reduction layers.

5. Conclusions

In this paper we have presented our approach on dialog act recognition on the DIHANA corpus using an end-to-end classifier to predict the labels for the three levels defined in the annotation scheme. This way, the relations between levels are captured implicitly, contrarily to what happened in our previous approach on the task, which used independent per-level classifiers. Additionally, we have used approaches on segment and context information representation which have recently been proved more appropriate for the task.

First, we have shown that character-based segment representation also performs better than word-based representation on the multi-label classification problems and that the combination of both approaches surpasses each individual approach. In this sense, on the combination of all levels, the combined approach surpassed the word- and character-level approaches by around four and one percentage points, respectively.

Then, we have shown that it is important to have per-level dimensionality reduction layers in order to specialize the segment representation for each level. Additionally, the performance is improved when a cue for the hierarchical relation between the levels is provided by considering the output for the upper levels when predicting the labels for each level.

Furthermore, we have shown that the relation between levels is also important when providing context information concerning the dialog history, as a combined summary of the classifications of the preceding segments led to better results than three independent per-level summaries.

Finally, by providing information concerning speaker changes and forcing the segments with Level 1 labels concerning dialog structuring or communication problems to have no labels on the remaining levels, we achieved 95.67% accuracy on the combination of the three levels, which is over three percentage points above our previous approach and even surpasses the results achieved on the simplified single-label classification problem approached by previous studies.

As future work, we intend to explore how our approach can be adapted to perform automatic segmentation of the turns instead of relying on a priori segmentation and assess the impact on the overall dialog act recognition performance.

6. Acknowledgements

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.
7. References


Abad, Alberto, 281
Aguilar, Lourdes, 112
Aksnes, Mari, 172
Alba-Castro, José Luis, 204
Alvarez, Aitor, 102, 267
Aliás Pujol, Francesc, 132
Antón, Josu, 68
Antón-Martín, Javier, 262
Arnela, Marc, 132
Arzelus, Haritz, 102, 267

Barbany, Oriol, 30
Barras, Claude, 194, 211
Batista, Cassio, 77
Benderitter, Ian, 72
Bernath, Conrad, 102, 267
Bonafonte, Antonio, 20, 30, 117, 152, 170
Borbély, Gábor, 286
Bordel, Germán, 249
Bredin, Hervé, 194, 211
Burga, Alicia, 40

Cabello, Maria, 245
Calvo de Lara, José R., 227
Calvo, Jose R., 254
Campbell, Edward L., 227
Cannings, Nigel, 231, 272
Cardeñoso-Payo, Valentín, 97, 112, 157, 161

Carrilero, Mikel, 68
Castan, Diego, 208
Castro-Bleda, Maria Jose, 286
Chollet, Gérard, 172, 231, 272
Codina, Marta, 291
Cordasco, Gennaro, 172
Corrales-Astorgano, Mario, 112
Cross Vila, Laura, 60

Cunha, Conceicao, 137
Ciscar, Vicent Andreu, 257
D’Haro, Luis Fernando, 55
Darna-Sequeiros, Javier, 64
de Benito, Diego, 224
de Córdoba, Ricardo, 55
de Velasco, Mikel, 68
Dehak, Najim, 236
Delgado, Héctor, 194, 211
Deroo, Olivier, 172
Dias, Ana Larissa, 77
Docío-Fernández, Laura, 35, 82, 204, 240, 277
Doetsch, Patrick, 257
Dominguez, Monica, 40
Dugan, Nazim, 231, 272

Escolano, Carlos, 60
Escudero-Mancebo, David, 97, 112, 157, 161
Esposito, Anna, 172
Espín, Juan Manuel, 159
Eszter, Iklódi, 286
Evans, Nicholas, 194, 211

Farrús, Mireia, 20, 40
Fernández-Ruanova, Begoña, 172
Ferreiros, Javier, 55
Fischer, Volker, 216
Fodge, Kimber, 291
Fonollosa, José A. R., 60
Font, Roberto, 159
Frahm, Jens, 137
Freixes, Marc, 132

Gallardo-Antolín, Ascensión, 15
<table>
<thead>
<tr>
<th>Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>García, Eneritz</td>
<td>267</td>
</tr>
<tr>
<td>García-Mateo, Carmen</td>
<td>35, 72, 277</td>
</tr>
<tr>
<td>García-Perera, L. Paola</td>
<td>236</td>
</tr>
<tr>
<td>Garrido, Juan-María</td>
<td>291</td>
</tr>
<tr>
<td>Gahabí, Omid</td>
<td>184, 216</td>
</tr>
<tr>
<td>Gilbert, James M.</td>
<td>163</td>
</tr>
<tr>
<td>Gimeno, Pablo</td>
<td>87, 220</td>
</tr>
<tr>
<td>Giménez, Adrià</td>
<td>257</td>
</tr>
<tr>
<td>Glackin, Cornelius</td>
<td>172, 231, 272</td>
</tr>
<tr>
<td>Golik, Pavel</td>
<td>257</td>
</tr>
<tr>
<td>Gomez Suárez, Javier</td>
<td>166</td>
</tr>
<tr>
<td>Gomez, Angel M.</td>
<td>45, 142</td>
</tr>
<tr>
<td>Gomez-Alanis, Alejandro</td>
<td>45</td>
</tr>
<tr>
<td>Gonzalez-Dominguez, Javier</td>
<td>179</td>
</tr>
<tr>
<td>Gonzalez-Rodriguez, Joaquín</td>
<td>179</td>
</tr>
<tr>
<td>González López, José Andrés</td>
<td>45, 117, 163</td>
</tr>
<tr>
<td>Gonzalez-Ferrer, César</td>
<td>112</td>
</tr>
<tr>
<td>Gordeeva, Olga</td>
<td>172</td>
</tr>
<tr>
<td>Granell, Emilio</td>
<td>92, 174, 267</td>
</tr>
<tr>
<td>Green, Phil D.</td>
<td>163</td>
</tr>
<tr>
<td>Guasch, Oriol</td>
<td>132</td>
</tr>
<tr>
<td>Guinaudeau, Camille</td>
<td>194</td>
</tr>
<tr>
<td>Gully, Amelia</td>
<td>163</td>
</tr>
<tr>
<td>Hernaez, Inma</td>
<td>50, 107, 122, 166</td>
</tr>
<tr>
<td>Hernandez, Gabriel</td>
<td>227</td>
</tr>
<tr>
<td>Hernando, Javier</td>
<td>10, 199</td>
</tr>
<tr>
<td>Hernández-Gómez, Luis A.</td>
<td>262</td>
</tr>
<tr>
<td>Huang, Zili</td>
<td>236</td>
</tr>
<tr>
<td>India Massana, Miquel Angel</td>
<td>199</td>
</tr>
<tr>
<td>Inglés-Romero, Juan Francisco</td>
<td>159</td>
</tr>
<tr>
<td>Jauk, Igor</td>
<td>170, 189</td>
</tr>
<tr>
<td>Jorge, Javier</td>
<td>257</td>
</tr>
<tr>
<td>Joseph, Arun</td>
<td>137</td>
</tr>
<tr>
<td>Juan, Alfons</td>
<td>257</td>
</tr>
<tr>
<td>Justo, Raquel</td>
<td>68, 172</td>
</tr>
<tr>
<td>Khan, Umair</td>
<td>10</td>
</tr>
<tr>
<td>Khosravani, Abbas</td>
<td>231</td>
</tr>
<tr>
<td>Kimura, Takuya</td>
<td>97</td>
</tr>
<tr>
<td>Kyslitska, Daria</td>
<td>172</td>
</tr>
<tr>
<td>Külebi, Baybars</td>
<td>25</td>
</tr>
<tr>
<td>Labrador, Beltran</td>
<td>224</td>
</tr>
<tr>
<td>Lindner, Fred</td>
<td>172</td>
</tr>
<tr>
<td>Lleida, Eduardo</td>
<td>1, 6, 87, 220, 296</td>
</tr>
<tr>
<td>Llombart, Jorge</td>
<td>296</td>
</tr>
<tr>
<td>Lozano-Diez, Alicia</td>
<td>179, 224</td>
</tr>
<tr>
<td>López Otero, Paula</td>
<td>82, 240</td>
</tr>
<tr>
<td>López-Espejo, Iván</td>
<td>142</td>
</tr>
<tr>
<td>López-Gonzalo, Eduardo</td>
<td>262</td>
</tr>
<tr>
<td>Machuca, María J.</td>
<td>97</td>
</tr>
<tr>
<td>Martínez Hinaresjos, Carlos David</td>
<td>92, 127, 174, 267</td>
</tr>
<tr>
<td>Martins de Matos, David</td>
<td>281, 301</td>
</tr>
<tr>
<td>Martins, Paula</td>
<td>137</td>
</tr>
<tr>
<td>Martín-Doñas, Juan M.</td>
<td>142</td>
</tr>
<tr>
<td>Martínez, Carlos David</td>
<td>102</td>
</tr>
<tr>
<td>Martínez-Castilla, Pastora</td>
<td>112</td>
</tr>
<tr>
<td>Martínez-Villaronga, Adrià</td>
<td>257</td>
</tr>
<tr>
<td>Maurice, Benjamin</td>
<td>194</td>
</tr>
<tr>
<td>McLaren, Mitchel</td>
<td>208</td>
</tr>
<tr>
<td>Miguel, Antonio</td>
<td>1, 6, 87, 220, 296</td>
</tr>
<tr>
<td>Mingote, Victoria</td>
<td>1</td>
</tr>
<tr>
<td>Montalvo, Ana R.</td>
<td>254</td>
</tr>
<tr>
<td>Montenegro, César</td>
<td>172</td>
</tr>
<tr>
<td>Moreno, Asuncion</td>
<td>170</td>
</tr>
<tr>
<td>Morros, Josep Ramon</td>
<td>199</td>
</tr>
<tr>
<td>Municio Martín, Jose Antonio</td>
<td>166</td>
</tr>
<tr>
<td>Murphy, Damian</td>
<td>163</td>
</tr>
<tr>
<td>Nandwana, Mahesh Kumar</td>
<td>208</td>
</tr>
<tr>
<td>Navas, Eva</td>
<td>50, 107, 122, 147, 166</td>
</tr>
<tr>
<td>Ney, Hermann</td>
<td>257</td>
</tr>
<tr>
<td>Odriozola, Igor</td>
<td>50</td>
</tr>
<tr>
<td>Oliveira, Catarina</td>
<td>137</td>
</tr>
<tr>
<td>Ortega, Alfonso</td>
<td>1, 6, 87, 220, 296</td>
</tr>
<tr>
<td>Palau, Ponç</td>
<td>199</td>
</tr>
<tr>
<td>Parcheta, Zuzanna</td>
<td>127</td>
</tr>
<tr>
<td>Pascual, Santiago</td>
<td>30, 117, 152</td>
</tr>
<tr>
<td>Patino, Jose</td>
<td>194, 211</td>
</tr>
<tr>
<td>Pavão Martins, Isabel</td>
<td>281</td>
</tr>
<tr>
<td>Peinado, Antonio M.</td>
<td>45, 142</td>
</tr>
<tr>
<td>Peláez-Moreno, Carmen</td>
<td>15</td>
</tr>
<tr>
<td>Pereira, Víctor</td>
<td>170</td>
</tr>
<tr>
<td>Periero-Codosero, Juan M.</td>
<td>262</td>
</tr>
<tr>
<td>Peñagarikano, Mikel</td>
<td>249</td>
</tr>
<tr>
<td>Piñeiro-Martín, Andrés</td>
<td>35</td>
</tr>
<tr>
<td>Pompili, Anna</td>
<td>281</td>
</tr>
</tbody>
</table>
Povey, Daniel, 236
R. Costa-Jussà, Marta, 60
Raman, Sneha, 107, 122
Ramírez, Jose M., 254
Ramírez, Pablo, 224
Ramos-Muguerza, Eduardo, 204
Recski, Gábor, 286
Reiner, Miriam, 172
Ribeiro, Eugénio, 301
Ribeiro, Ricardo, 301
Rituerto-González, Esther, 15
Roble, Alejandro, 254
Rodríguez-Fuentes, Luis J., 249
Romero, Verónica, 92, 174
Ríos, Antonio, 97
S. Kornes, Maria, 172
Safari, Pooyan, 10
Sagastiberri, Itziar, 199
Salamea, Christian, 55
Sampaio Neto, Nelson, 77
San-Segundo, Rubén, 55
Sanchez, Jon, 50
Sanchis, Albert, 257
Santana, Riberto, 172
Sarasola, Xabier, 122, 147
Saraxaga, Ibon, 122, 147
Sayrol, Elisa, 199
Schlögl, Stephan, 172
Serrano, Luis, 50, 107, 122, 147
Serrà, Joan, 117, 152
Silva, Samuel, 137
Silvestre-Cerdà, Joan Albert, 257
Socoró, Joan Claudi, 132
T. Toledano, Doroteo, 64, 224, 245
Tapias Merino, Daniel, 262
Tarrés, Laia, 170
Tavarez, David, 122, 147
Teixeira, António, 137
Tejedor, Javier, 245
Tejedor-García, Cristian, 97, 157
Tenorio-Laranga, Jofre, 172
Tilves Santiago, Darío, 72
Torres, M. Inés, 68, 172
Varona, Amparo, 249
Vicente-Chicote, Cristina, 159
Villalba, Jesús, 236
Viñals, Ignacio, 6, 87, 220
Wanner, Leo, 40
Yin, Ruiqing, 194, 211
Öktem, Alp, 20, 25