Speech recognition based strategies for on-line Computer Assisted Language Learning (CALL) systems in Basque

Igor Odriozola Sustaeta

Aholab Signal Processing Laboratory
Department of Communication Engineering

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Advisors: Inma Hernáez and Eva Navas

A dissertation submitted to the University of the Basque Country for the degree of Doctor of Telecommunication Engineering

March 13, 2019

Speech recognition based strategies for on-line Computer Assisted Language Learning (CALL) systems in Basque

Igor Odriozola Sustaeta

Aholab Signal Processing Laboratory
Department of Communication Engineering

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Advisors: Inma Hernáez and Eva Navas

A dissertation submitted to the University of the Basque Country
for the degree of Doctor of Telecommunication Engineering

March 13, 2019

Lengoagetan ohi inçan
Estimatze gutitan
Oray aldiz hic behar duc
Ohoria orotan.

Heuscara
Habil mundu gucira

Bernart Etxepare, 1545

I would like to dedicate this thesis to my mother, who transmitted me Basque language from her heart, and to my father, who aroused his passion for science in me.

Abstract

There is a growing interest in the use of speech technology in computer-assisted language learning (CALL) systems. This is mainly due to the fact that speech technologies have considerably improved during the last years, and nowadays more and more people make use of them, ever more naturally. Literature shows that two major points of interest are involved in the use of Automatic Speech Recognition (ASR) in CALL: Computer-assisted Pronunciation Training (CAPT) and Spoken Grammar Practise (SGP). In a CAPT typical application, the user is required to read and record a sentence, and send it to the learning application server. It returns a result, generally using a three-level colour system indicating which phone has been correctly pronounced and which has not. SGP applications are not very popular yet, but some examples are multiple-choice tests, where users orally choose an answer between several choices, or even the Intelligent Language Tutoring Systems, where learners respond to cues given by the computer and the system provides feedback. Such tools can be used to strengthen the students' autonomy on their learning process, giving the opportunity of using voice to improve their pronunciation or to do grammar exercises outside the classroom.

In this work, two applications have been considered: on the one hand, a classical CAPT system, where the student records a predefined sentence and the system gives a feedback about the pronunciation; on the other hand, a novel Word-by-Word Sentence Verification (WWSV) system, where a sentence is verified sequentially, word by word, and as soon as a word is detected it is displayed to the user. The WWSV tool gives the option of creating a tool to solve grammar exercises orally (SGP). Both systems rely on utterance verification techniques, as the popular Goodness-of-pronunciation (GOP) score.

The acoustic database chosen to train the systems is the *Basque Speecon-like* database, the only one publicly available for Basque, specifically designed for speech recognition and recorded through microphone. This database presents several drawbacks, such as the lack of a pronunciation lexicon and some annotation files. Furthermore, it contains much dialectal speech, mainly in the free spontaneous part. In ASR, phonetic variations can be modelled using a single acoustic model. However, CAPT systems require "clean" models to use as reference. Thus, some work had to be carried out on the annotation files of the database. A number of transcriptions have been changed and a new lexicon

has been created considering different dialectal alternatives. It is noticeable that the new lexicon contains in average 4.12 different pronunciations per word.

The speech recognition software used in this thesis is *AhoSR*. It has been created and developed by the *Aholab* research group, and it has been designed to cope with different recognitions tasks. In this thesis, utterance verification techniques have been implemented to be run together with the basic tasks. To do so, a parallel graph has been implemented to obtain GOP scores. For CAPT and WWSV tasks, specific search graphs have been added, in order to adapt to the needs of each of them. In addition, sockets have been implemented in the audio-input module of *AhoSR*. This allows real time performing when accessing the recogniser through the internet, and so it gives us the opportunity for *AhoSR* to be installed on a server, with universal access.

Different ways to train Hidden Markov Models (HMM) have been analysed in depth. Initially, HMMs of better quality were expected by means of using the new dictionary with alternatives. However, results do not show this, probably because of the big amount of alternative pronunciations. The addition of some manually corrected data (15 % of the training set) allows obtaining similar results to those obtained using a single-entry dictionary. In order to take advantage of the manually corrected transcriptions, different ways of training HMMs have been analysed. Thus, we have found that slightly better HMMs are achieved using data with few transcription errors in the initial stages of the training and then using the whole database.

To build the initial system, two GOP distributions were considered necessary to classify each phone: the distribution of the correctly pronounced phones and the distribution of the incorrectly pronounced ones. The GOPs of the incorrectly pronounced phones were obtained simulating errors and obtaining the GOP scores by forced alignment in the *AhoSR* decoder. Thus, the thresholds between correctly and incorrectly uttered phones were calculated as the Equal Error Rate (EER) point of both distributions. This approach was implemented in an initial prototype, and several laboratory experiments were performed which produced very good results. Then, the system was tested in more realistic environments: Basque language schools, among 20 students. The objective results along with the survey filled in by the 20 students who tested the system were really promising.

The initial prototype was executed locally, and we felt the need of developing a more universal system in order to be accessed from any device and anywhere. Thus, we took advantage of the specifications of the recent HTML5 standard, which let the browser access the audio input, regardless of the platform, by means of the *audio API*. This has given us the opportunity to create a system accessible from any operative system. Moreover, for the WWSV-based SGP task, another API of HTML5 has been used (the *web API*), which creates socket-like connections between the browser and the server, in order to send audio data on the fly.

Several drawbacks have been managed for the on-line implementation of the system: for example, due to the different devices that users will use to pick up audio, some kind of parameter normalisation is needed. Furthermore, an on-line normalisation technique is necessary, since in WWSV continuous feedback must be provided before the whole signal has arrived to the recogniser. Different techniques have been tested to implement Cepstral Mean and Variance Normalisation (CMVN) and estimate the initial values of cepstral means and variances. The best results have been obtained by a hybrid approach proposed in this work, so that the initial means are estimated using the first N frames, and initial variances are obtained from the training datasets.

In addition, a new CMVN technique has been devised in this thesis: the Multi-Normalisation Scoring (MNS) based CMVN. MNS consists in generating multiple observation likelihood scores by normalising the incoming Mel-Frequency Cepstral Coefficients (MFCC) using means and variances computed from different speech datasets recorded under different conditions. The MNS-based CMVN consists in computing the probabilities of a frame to belong to different training datasets; thus, these probabilities can be used as weights to calculate an estimation of the actual means and variances. The results obtained are remarkable, mainly for clean signals. The greatest advantage of using MNS is that the CMVN can perform on-line, frame by frame, with no need to analyse the neighbouring frames or the frames of a segment to which it belongs.

Using the same MSN method, a novel and effective on-line Voice Activity Detector (VAD) has been devised as well. In a validation experiment, comparing the results of our MNS-based VAD with the results of two ITU-T VAD algorithms (G.720.1 and G.729b), we have obtained better overall results, since the classification errors are considerably lower for non-speech frames, and are comparable for speech frames. This makes our system useful for systems that require low speech error rates and also for low non-speech error rates.

Finally, Neural Networks have been used in an attempt to see the impact of different parameters at the time of training a classifier. As a consequence, we have seen that GOP scores are the most efficient parameters among durations and log-likelihoods of the previous, current and posterior phones. The results of the experiments are coherent with those obtained in the initial system.

Acknowledgements

I thank my thesis directors for giving me the opportunity to write this dissertation at the *Aholab* research group in UPV/EHU (University of the Basque Country), for their professional advice, and for the confidence invested in me.

I thank in particular Luis Serrano, my co-worker who readily has helped me so often, and all our colleagues from the research group, for always being available to lend a hand.

I would also like to thank all people who supported me in writing this thesis, and those who have missed me during the time in which I have been working on this.

Declaration

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been acknowledged.

Abbreviations

ASR	Automatic Speech Recognition
CALL	Computer Assisted Language Learning
CAPT	Computer Assisted Pronunciation Teaching
CMVN	Cepstral Mean and Variance Normalisation
CF	Corrective Feedback
GMM	Gaussian Mixture Model
GOP	Goodness Of Pronunciation
HMM	Hidden Markov Model
L1, L2	First language (Mother tongue), Second language (target language)
MFCC	Mel Frequency Cepstral Coefficient
MLP	Multi-Layer Perceptron
MNS	Multi-Normalisation Scoring
NN	Neural Network
PER	Phone Error Rate
SGP	Spoken Grammar Practice
VAD	Voice Activity Detection
WWSV	Word-by-Word Sentence Verification

Contents

List of Figures

List of Tables

PART I

Introduction

CHAPTER 1

Introduction

1.1 Prologue

I feel very fortunate for having had the opportunity of developing this thesis dissertation. It unites my two greatest passions: Basque language and science. Not many people can say that they are working in something they are passionate about, and in my case it happens twice over.

I started teaching Basque language lessons while I was studying telecommunication engineering, probably because I have always been attracted by both areas, and I knew immediately that my professional future would have to join them somehow. *Aholab*, the laboratory to which I belong, gave me the opportunity for this.

The laboratory opened up a new research line within its areas of interest, with a project based on implementing Automatic Speech Recognition (ASR) technology in a second language learning commercial software (for Basque), with the aim of improving the quality of the communication between the learner and the computer, and make it easier for the learner to take advantage of his or her learning process autonomously. This was my first project in *Aholab* and this is how I thereby started working in the field of ASR.

Just after obtaining the Master's degree in *Language Analysis and Processing*, I built the first version of what today is *AhoSR*, our Automatic Speech Recognition (ASR) system for Basque. It was a word grammar based decoder, written in C++ for Windows, and it accepted wav files as well as direct audio as input. On the basis of that initial prototype, *AhoSR* started growing and improving.

Several years have now passed since then, and that initial recogniser has become a more complete and stable software. Currently *AhoSR* can also be used for other tasks such as phonetic recognition, utterance verification, and large vocabulary continuous speech recognition (LVCSR). It can be accessed via web (using HTML5 specifications), thus providing potential users with many accessibility opportunities to be used remotely, regardless of the platform used. Some demos can be found in: `http://aholab.ehu.eus/users/igor/demos.html`.

1.2 Outline

As technology develops, new opportunities arise which can be applied to language learning. The technologies that have emerged over the last years have had a wide impact on the field of Computer Assisted Language Learning (CALL), not only in practice, but also in its discourse, research and pedagogies. The advent of the Internet (including the current HTML5 web specifications) along with the more recent mobile platforms (such as smart-phones and tablets) has been a global revolution, and this has also had a big impact on the way in which language classes are conducted and organised. Some experts in the field believe that the development of these new technologies gives rise to completely new ways of communicating and thinking as well, and, in their view, it may bring a complete reassessment of the pedagogy used in the field of language learning[1].

According to [2], the items concerning the field of CALL which have attracted the greatest attention in the literature and have appeared to have the greatest impact in CALL practice and research are the following: authoring software, learning management systems (LMS), audio- and video-conferencing, artificial intelligence (AI) and intelligent systems, mobile technologies, and speech recognition and pronunciation-training technologies. Speech technologies —more precisely, ASR and computer-assisted pronunciation training (CAPT)— are therefore considered as a point of interest in the current field of CALL.

Applications of ASR are diverse. One such usage is in Intelligent Language Tutoring Systems (ILTS), where learners respond to cues given them by the computer. The oral output uttered by the learners is then picked up, errors are located, and feedback is provided in response to these errors. The other major use of speech technologies is CAPT. The training can be carried out considering segmental aspects (individual phoneme sounds) or supra-segmental aspects (prosodic features). Most applications implementing CAPT technologies rely on visual representations of the sounds produced by the learners compared with a correct model. These technologies are increasingly used in other areas such as speech pathology detection and treatment.

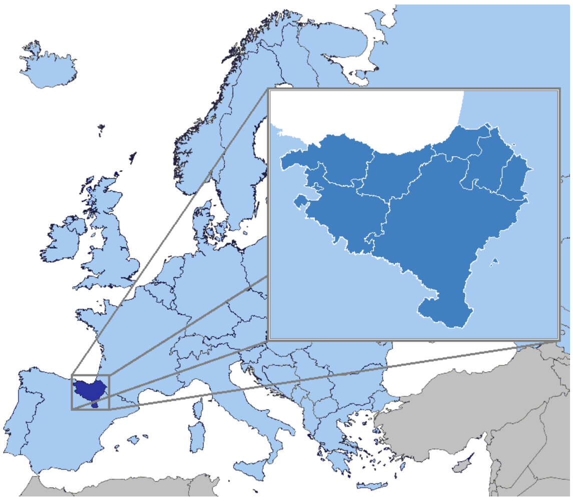
Developments in web-technology have enabled new interfaces previously not possible. HTML5 provides brand new possibilities of using the Internet with voice-response applications, through its *web audio API*. By means of HTML5, web-based voice input applications can be linked to server-based ASR and pronunciation training applications, expanding thus opportunities for self-study as well as links with other web-based technologies, such as Learning Management Systems (LMS) or other authoring tools. HTML5 is expected to substitute Adobe Flash applications and Java applets in the short term, since it allows web-browsers to manage audio natively. This means that HTML5 technology is cross-platform, only dependant of the browser. Currently, the most popular browsers have it already implemented.

This thesis describes a strategy of how to implement both an ASR-based CALL system and a CAPT system for Basque using a standard ASR database. The problems encountered for the on-line web-implementation of the system will also be explained, as well as the proposed methods to solve them.

1.2.1 Thesis motivation

Basque language is a European language isolate, which is believed to be one of the few surviving pre-Indo-European languages in Europe, and the only one in Western Europe. The language's origins are not conclusively known, though the most accepted current theory is that early forms of Basque developed prior to the arrival of Indo-European languages in the area, including the Romance languages that geographically surround the Basque-speaking region, known as *Euskal Herria* (Basque Country).

Euskal Herria is administratively divided into two: *Ipar Euskal Herria* (Northern Basque Country), under French administration, and *Hego Euskal Herria* (Southern Basque Country), under Spanish administration (see Figure 1.1) . Ipar Euskal Herria corresponds to about half of the *Département des Pyrénées Atlantiques*, and Basque does not have official recognition there, since today the sole official language of the French Republic is French. Hego Euskal Herria is also divided into two: the Autonomous Basque Community and Nafarroa (the former Kingdom of Navarre). Basque is co-official with Spanish in the Autonomous Basque Community, and in Nafarroa it has a heterogeneous status, which depends on the region. Obviously, considering this diglossia situation in which the official languages have a prestige status compared to the non-official one, Spanish and French exert a significant influence on Basque. According to a Basque Government survey of 2012, the Basque language is spoken by 27% of Basques in Euskal Herria (714 136 out of 2 648 998). Of these speakers, 663 036 belong to the Southern Basque Country and the remaining 51 100 belong to the Northern Basque Country [3].



(image taken from Wikipedia)

Fig. 1.1: Location of *Euskal Herria* in Europe

The figures reflect that the use of the Basque language is far from being normalised. Many people are nowadays learning Basque. Autonomic and local administrations provide financial support, under specific conditions, to the people who want to learn Basque. Basque teaching has significantly evolved over the past 40 years, and the language schools are regulated by an organism dependent of the Basque Government. The field of Basque language learning deserves the same technological opportunities as the languages that are most spoken.

Basque language has a weak point. The Standard Basque (or *Euskara Batua*) was not developed until the late 1960's, which means that *Euskara Batua* is still nobody's "real" native language. Nowadays, it is used in education at all levels, from elementary school to the university, on television and radio, an in the vast majority of all written production in Basque [4]. Nevertheless, it was primarily created for written purpose. An indicator of this is the fact that the Royal Academy of the Basque Language (*Euskaltzaindia*) has published its first exhaustive work about pronunciation and prosody of *Euskara Batua* in 2014 [5], almost 50 years after its first steps. Over this time, Basque teachers have been teaching a language that was not completely defined, and the learners, the real speakers of Standard Basque, have shaped the spoken *Euskara Batua*.

As mentioned above, *Euskara Batua* is still nobody's "real" native language. Native speakers use their corresponding dialectal variety, and even though *Euskara Batua* is taught at schools, it is seen as a very formal and even artificial version. Furthermore, Basque is a language of considerable dialectal diversity. The division of the Basque-speaking area into geographical dialects and sub-dialects has been a traditional concern of researchers on the Basque language. However, the fact is that virtually every town or village speaks its own variety [6]. This gives an idea of the complexity that Basque teachers have to cope with when teaching Basque in the classroom.

Speech technologies in CALL systems could be very helpful in this normalisation situation of Basque. The tools proposed in this thesis can be very helpful for Basque language learners in their way to speak as native Basques do and achieve a spoken Basque of quality, since they can use voice not only to assess their pronunciation, but also to resolve grammar exercises. This would entail an improvement on their oral production skills. Additionally, the learners could also feel more motivated with such tools, especially those who do not have opportunities to practise the target language outside classroom.

1.2.2 Goals

The overall objective of this thesis dissertation is to analyse ASR-based strategies to be implemented in CALL applications designed to learn Basque. To that end, utterance verification techniques have been considered, computed over the recognition decoder, since they offer two potential advantages: on the one hand, the possibility to assess the pronunciation of an utterance (CAPT systems), taking as a reference the speech contained in the *Basque Speecon-like* acoustic database. On the other hand, the possibility of having a tool that verifies the response of the user word by word in real time, which could be useful to individually practise grammar doing exercises orally. Such technique has been called Word-by-Word Sentence Verification (WWSV) in this work, and this term will be used from now on when referring to such systems.

To reach this goal and ensure the usefulness of the resulting conclusions, the following partial objectives need to be also considered:

- Obtaining good quality acoustic models from a database designed for standard ASR purposes, in order to use them to train confidence scores at sub-word level for phone verification.

- Creation and improvement of decision-thresholds for the confidence scores. This involves analysing different ways of grouping phones.
- Analysing the problems that an on-line implementation of such a system has, as well as proposing a design solution.
- Studying and applying other signal processing techniques to improve the overall performance of the systems, such as Cepstral Mean and Variance Normalisation (CMVN) and Voice Activity Detection (VAD).

1.2.3 Thesis structure

This thesis dissertation is divided into 4 parts and contains 11 chapters.

Part I is the introduction, and consists of 2 chapters. The first chapter is the current one (Chapter 1). The second chapter, Chapter 2, provides a summary of the literature published in the field of speech technologies for CALL systems. We will focus on the use of ASR in such applications, as well as on different techniques used for pronunciation scoring. Finally, some conclusions are set out.

Part II introduces the basis of the initial system. Chapter 3 describes the acoustic database used in this thesis and the modifications made to it in order to use it to obtain good quality acoustic models. A detailed description of the database is provided here, as well as some work made on the database to improve annotation files and create a lexicon.

Chapter 4 shows the structure and functionalities of *AhoSR*, the underlying speech recognition software used in this thesis, created and developed by the *Aholab* research group. The modifications and adaptations made on *AhoSR* to implement the use of verification scores and the on-line word-by-word verification are also described in this chapter.

In Chapter 5, an exhaustive analysis of different ways of training acoustic models (HMMs) is given. The database used in this thesis has several drawbacks, due to the big amount of dialectal variations it contains. The creation of a lexicon with dialectal variations as alternatives is described, and also the effects of using such a lexicon to train Hidden Markov Models (HMM). Furthermore, a study of the use of different subsets for training has been carried out, even using different subsets at different stages of the training process. Phonetic recognition results of thus obtained HMMs are shown, and some conclusions are presented.

One of the main drawbacks to set pronunciation thresholds is the lack of incorrectly pronounced data. Chapter 6 shows the procedure followed in this thesis to overcome this issue, and different experiments to validate it. An application created to evaluate the procedure of obtaining thresholds in a realistic environment is also presented here. This application was designed to orally solve grammar exercises (using the WWSF system) and runs locally in each computer. The details of the evaluation with real students and the results obtained in those evaluations are finally described.

Part III describes different system improvements developed on the basis of the initial system. In Chapter 7, client/server implementation issues are described, making use of functionalities of the recent HTML5 specifications (the audio API). Besides, the websocket API is also used to send binary audio data to a *Nodejs* server and obtain feedback.

Chapter 8 introduces a novel method to detect speech segments in audio files, based on Multi-Normalisation Scoring (MNS). It is based on the observation likelihoods generated by the Gaussian Mixture Model (GMM) from the central-state of the silence HMM trained applying cepstral normalisation. An experiment has been carried out to compare the results obtained using this technique with those obtained with the most popular systems. Results are very competitive at different noise levels.

Chapter 7 introduces another novel technique to normalise cepstra on-the-fly, based on the MNS technique as well. This is one of the problems that arise when implementing the WWSV system in a server. Different strategies to cope with on-line cepstral normalisation are described, and some conclusions are reached.

In Chapter 10, the concept of incorrectly pronounced phoneme is revisited. Based on this, different Neural Networks are trained to be used as a classifier for a CAPT system, using different parameter sets in order to see the impact of each of them. The results of the experiments and some conclusions are finally drawn.

Part IV shows a summary and an outlook. Finally, Chapter 11 contains the general conclusions derived from this thesis, and some reflection about future research work.

CHAPTER 2

ASR technology in CALL systems

2.1 Introduction

Computer-assisted language learning (CALL) is a specialised field in applied linguistics concerned with the use of technology in language teaching and learning. In its initial stages, during the sixties, CALL mainly focused on drill-and-practice oriented software, within the framework of behaviourism and the audio-lingual approach to teaching languages. However, with the widespread availability and accessibility of personal computers and multimedia, and the advent of the Internet at the end of the 20th century, CALL diversified, expanded and evolved [7].

CALL in its early stages was seen as a largely subordinate area of English Language Teaching (ELT) [8], with books on second language acquisition and pedagogy devoting little space on the use of technology [9]. Today, the field of CALL is formed by several professional organisations which promote the use of technology —for example, *CALICO*, *EUROCALL* and *IALLT*—, and specialised journals devoted exclusively to it —such as *Language Learning & Technology*, *ReCALL*, *CALL*, *Journal of Computer-Mediated Communication* and *CALICO Journal*—. CALL is thus a well-established field with an extensive research agenda and diverse practice applications across all areas of second language acquisition.

Practice is at the heart of CALL, and computers have been used in all areas of language learning. They are considered particularly useful for the teaching of grammar, vocabulary, reading and writing, listening and pronunciation, and they have been widely used in language assessment. Many authors consider technology as a necessary item inside and outside the classroom, although always within a framework that takes account for the corresponding pedagogical and language acquisition principles. A proper supervision of the teacher about the use of technology outside the classroom is supposed to help learners have a better experience [10].

Nowadays self-access centres offer a new autonomous way to learn a second language. They have been enthusiastic consumers of educational technologies, and so self-access learning has become synonymous with technology-based learning. Within the field of

CALL, especially, autonomy has become an important issue. The advantages, disadvantages and consequences of autonomous learning are nowadays being analysed in an exhaustive manner. As in the case of self-access, however, researchers on autonomy emphasise that learners who engage in technology-based learning do not necessarily become more autonomous as a result of their efforts. A great deal depends on the nature of the technology and the use that is made of it [11]. This point of view is also shared by [12], which explain that the incorporation of technology in L2 learning has to adapt to the diverse needs of both learners and teachers, and it should focus on the process of meaning making and learning with technology, and not just a set of post-test scores.

Today, CALL not only makes use of specialised and sophisticated multimedia software, but also numerous web-resources, Web 2.0 tools and social networking software, learning management systems (LMS) and instruction tools, and mobile technologies. These tools are used in various degrees both inside and outside of the classroom for language learning and teaching purposes. This application diversity of CALL has given rise to a number of terms used to refer to the use of technological tools in the language learning classroom, such as:

- NBLT: Network-Based Language Teaching [13]
- CMC: Computer Mediated Communication [14]
- WELL: Web-Enhanced Language Learning [15]
- MALL: Mobile Assisted Language Learning [16]
- ICALL: Intelligent CALL [17]
- CALT: Computer Assisted Language Testing [18]
- eLearning: Learning with Technology [19]

Despite the wide range of CALL applications and the equally wide list of acronyms in the field, 'CALL' remains the umbrella term for the use of technology in language teaching and research [1]. However, nowadays the term CALL is evolving to ICALL or Intelligent CALL. ICALL integrates natural language processing (NLP) and artificial intelligence (AI) modelling into CALL, in order to improve the interactions between computers and users, and individualising learning experience.

2.2 ASR technology in CALL systems

Already in 1996, the author of [20] concluded from one of her user studies that only CALL programs that make use of the full potential of the computer, mainly by providing immediate and appropriate feedback, would produce higher learning results. Today, we can say that, due to the great improvement in ASR technology, we are closer to achieving a naturalistic communication between computers and users, and more appropriate feedback is provided.

CALL systems mostly address written production, but the beneficial characteristics of CALL can also apply for speaking practice [21]. Practice is important, since spoken production requires a higher cognitive load and control over the articulatory system [22]. To train control over cognitive load and articulation, CALL systems should allow for speaking practice and provide automatic corrective feedback (CF) on speaking performance. In [23], where a reflection about the effectiveness of CALL is presented, the author calls for further studying the impact of CALL in the “relatively unexplored” field of “speaking online”.

The advances in the field of ASR technology make it possible to implement focused exercises for spoken production [24]. In a review of 350 studies that report on the effectiveness of CALL systems, Golonka et al. [25] found evidence that CALL technology can impact pronunciation training (computer-assisted pronunciation training, CAPT) and that ASR is employed for spoken practice; however they do not report systems that provide automatic CF on Spoken Grammar Practice (SGP). A review study of CALL systems using ASR in [26] showed that the available systems addressed communicative skills or pronunciation, while at that time (2011) there were no systems that offered spoken practice with automatic CF on grammar errors. However, since accurate grammar is an important aspect of proficiency and, therefore, a main pedagogical goal in L2 learning, a system to practice grammar in the oral modality seems to be a valuable application for L2 learning.

Since the end of the 20th century, various systems based on ASR technology have been developed which provide practice and CF in L2 speaking. Many of them include CAPT, some implement spoken practice, and fewer make use of SGP. Some examples

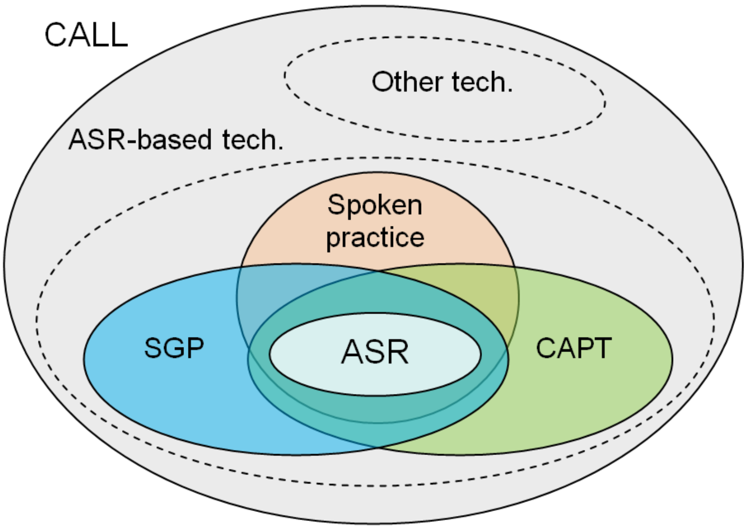


Figure 2.1: Place held by ASR technology within SGP and CAPT in the field of CALL, according to the type of technology used.

are: *FLUENCY* [27], the *Tactical Language Training System* [28], the *AzAR* system [29], the *SPELL* system [30][31], *Carnegie Speech Native Accent* [32], *Saybot* [33], *Euronounce* system [34], *EduSpeak* [35] and *Tell me More* bought in 2013 by *Rosetta Stone* (`www.rosettastone.com`, 2018). Currently, there are many mobile and tablet apps such as *Babble*, *Busuu*, *Mondly* and *Rocket Languages*. Many language-learning apps are currently coming in and out of the market.

Next subsections describe, in more detail, the state of the art in the areas of SGP and CAPT, the two ASR-based implementations chosen for this thesis (see Figure 2.1). While many research and publications have been devoted to CAPT in the last years, it seems that further research and resources are needed for the area of SGP.

2.2.1 Computer-assisted Pronunciation Training (CAPT)

Pronunciation errors can be divided into phonemic and prosodic error types [36]. Regarding the phonemic errors, the most important errors that a L2 student can do are substitutions, deletions and insertions. Less important errors are phonemes that are uttered but their sound is still different enough from a native speaker's pronunciation that it is noticeable that a speaker still has an accent. They can also be seen as substitutions, but they do not impair communication. Regarding prosodic errors, a non-native accent can be categorised is terms of stress, rhythm and intonation. All such errors are closely linked. All this makes the pronunciation a multi-dimensional problem that is difficult to solve with a single approach.

a) Phonemic errors

The first works on automatic pronunciation scoring were published at the beginning of the 1990s. They were mainly focussed at word-level and phrase-level, augmented by measures of intonation, stress and rhythm: the system described in [37] and the prototype developed under the *SPELL* project (Interactive System for Spoken European Language Training) [38] used signal processing techniques such as similarity metrics, spectral distance and differences in fundamental frequency and speech power, to rate the pronunciation quality of a word or sentence uttered by a student. In [39] HMMs were used to evaluate Dutch words, performing forced recognition and using native and non-native judged data. However, these systems typically required several recordings of native utterances to train the models for each word in the teaching material. They were therefore text-dependent with the disadvantage that the teaching material could not be adjusted without making additional recordings.

In those years, ASR with HMMs was also used to score complete sentences rather than smaller units of speech [40][41]. Systems aimed at teaching selected phonemic errors were also described in various papers: in [42] durational information of Viterbi decoding was employed. In [43] a *mispronunciation score* (MP) is used, which is the ratio between the likelihoods of non-native and native speech. In [44] three HMM-based scores are compared: log-likelihood score, log-posterior probability score and segment duration score, demonstrating that the log-posterior probability scores have the highest correlation with human ratings. Besides this HMM-based log-posterior probability based method, in [45] the log-likelihood ratio (LLR) between native-like and non-native models is adopted as the measure for mispronunciation detection. The results show that LLR based method has better overall performance than the posterior based method, but it needs to be trained with specific examples of the target non-native user population.

In 1999, CAPT became an international interest topic. That year, an entire issue of *CALICO* journal, one of the most important journals devoted to research and discussion on technology and language learning, was devoted to these technologies, and the first complete thesis about CAPT was presented by Witt [46], which introduced the *Goodness of Pronunciation* (GOP) score for phone-level pronunciation scoring [47], a variation of the posterior probability (see Chapter 6.2). From then on, this GOP measure has been widely used in pronunciation evaluation and mispronunciation detection tasks.

The basis of the design of the CAPT system proposed by Witt is shown in Figure 2.2. The front-end feature extraction converts the speech signal to a sequence of Mel-Frequency Cepstral Coefficients (MFCC), and these are used in two recognition passes: the first pass (the *forced alignment* pass) produces the acoustic segment boundaries and the corresponding triphone likelihoods, determined from the Viterbi alignment. In the second pass (the *phone loop* pass) each phone can follow the previous one with equal probability, and the log likelihoods are obtained over the segmentation provided by the first pass. Based on these results, the individual GOP scores are calculated for each

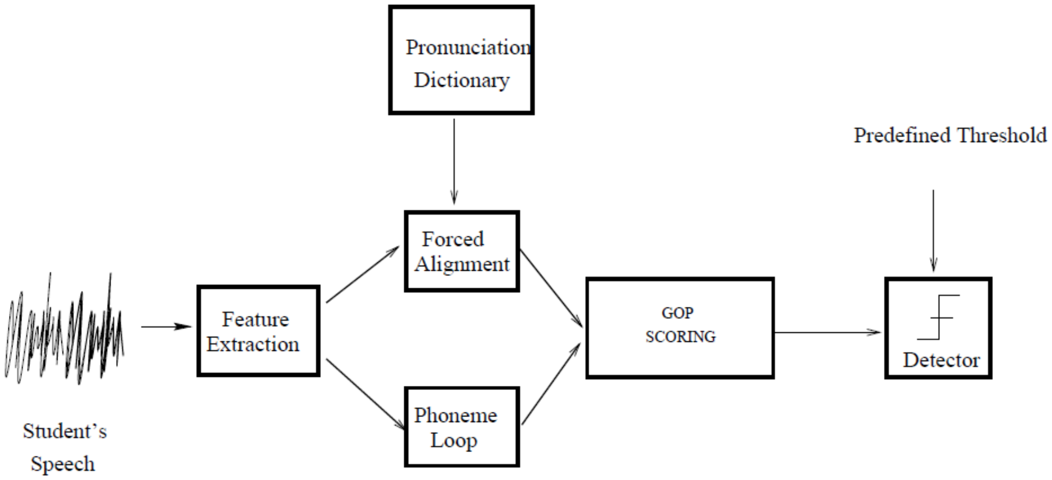


Figure 2.2: Block-diagram of the classic CAPT system described by Witt: phones whose scores are above the predefined threshold are assumed to be badly pronounced and are therefore rejected.

phone. Finally, a threshold is applied to each GOP score to reject badly pronounced phones. The choice of the thresholds depends on the level of strictness required. The selection of suitable thresholds is further discussed in Chapter 6.2.

Some variations of GOP score have also been proposed since then. In [48] a scaling log-posterior probability (SLPP) method is proposed for Mandarin mispronunciation detection, and a considerable performance improvement is achieved. In [49] the GOP-based method is combined with error pattern detectors for phone mispronunciation diagnosis with a serial and parallel structure, and they found that the serial structure can reduce the average error rate and improve diagnosis feedback. To improve the scores generated by the traditional GMM-HMM based speech recogniser, some discriminative training algorithms have been applied, e.g. Maximum Mutual Information Estimation (MMIE) [50], Minimum Classification Error (MCE) [51], and Minimum Phone Error (MPE) and Minimum Word Error (MWE) [52]. In [53], the acoustic models discriminatively refined by MPE are used for pronunciation proficiency evaluation, and in [54] they also investigate using MWE-trained HMM models to minimise mispronunciation detection errors for L2 English learners.

Classifier-based approaches have also been applied to mispronunciation detection, formulating it as a 2-class classification task. In [55] a decision tree based method is used to set thresholds for different kinds of mispronunciations achieving a significant improvement compared with a universal threshold. In [56] decision trees are also used along with Linear Discriminant Analysis (LDA) to distinguish different pronunciation errors of L2 learners of Dutch. Experimental results show that the classifier based approach has a good performance of detecting vowel pronunciation errors but a poor performance of detecting consonant pronunciation errors. It also shows that LDA yields a better detection performance than decision tree. Four different approaches are compared in [57]: GOP score, decision tree, and LDA with two kinds of features, i.e. acoustic–phonetic features and MFCCs. The results show that LDA based methods outperform the GOP and decision tree based methods. However, the tests are only carried out for two Dutch phones.

Support Vector Machine (SVM) classifiers were also used to improve the performance of CAPT systems by combining different features: confidence measures, phonetic features and MFCCs [58]. In [59], SVM is applied to classify the correct and incorrect pronunciations of Mandarin syllables by enhancing the discrimination of pronunciation variation with Pronunciation Space Models (PSM). Other research work, which implements SVM as the underlying classifier for mispronunciation detection in different scenarios, can be found in [60][61][62][63].

Recently, Deep Neural Network (DNN), which attempts to model high-level abstractions in data, has significantly improve the discrimination of acoustic models in ASR [64]. In [65], the authors explain that GOP scores estimated from DNN-based acoustic models correlate better with human expert's evaluations than conventional GOP scores obtained from conventional GMM-based system. An improved system is presented in [66], where DNN-trained acoustic models are used along with phone specific 2-class

Logistic Regression (LR) classifiers for English pronunciation quality scoring. Authors describe that this system outperforms the state-of-the-art SVM-based classifier, although not by much.

Deep Belief Networks (DBN) have also been used in [67] to mispronunciation detection and diagnosis in L2 English. The acoustic modelling is done using the hybrid DBN-HMM speech recognition framework, and a significant improvement on word pronunciation relative error rate was obtained on an L1 (Cantonese) dependent English learning corpus. However, it is computationally more expensive than the classic GMM-HMM based system.

b) Prosodic errors

Last years, there has been great interest in exploring automated methods to measure prosodic features of pronunciation. Prosodic features are those related to intonation, stress and fluency, such as the f_0 fundamental frequency contours (slope, average and maximum), the mean, maximum, minimum power per word, the distances between stressed and unstressed syllables, rate of speech, articulation rate, phonation time/ratio, mean phoneme duration etc.

f_0 contours have been widely used to provide a CF, as in *AzAR* system [29] and in *Euronounce* system [34]. In 2012, during the research stay in Germany to take the first steps for this thesis, an study was carried out to check the validity of the f_0 contour as a measure of the intonation [68]. In that work, RMSE (root-mean-square error) distances between $log(f_0)$ curves of a reference Basque speaker and 10 speakers (6 foreign and 2 Basque speakers, see Table 2.1) were calculated. Since the pitch perception of the human ear is proportional to the logarithm of frequency rather than to frequency itself, $log(f_0)$ curves were used. The results showed that the automatically computed RMSE distances between $log(f_0)$ curves are smaller for speakers of Basque language regardless

Table 2.1: RMSE of $log(f_0)$ curves of the speech of 8 speakers compared to the reference voice

	Mother tongue	Gender	Age	RMSE
01	Japanese	m	27	0.134
02	Macedonian	m	38	0.140
03	Amharic (Ethiopia)	m	32	0.134
04	German	m	42	0.165
05	Urdu (India)	m	31	0.129
06	Slovakian	f	26	0.135
07	Basque	f	35	0.111
08	Basque	m	34	0.113

of his/her gender (notice that typically the pitch of female voices is at around 200 Hz and the pitch of male voices at around 100 Hz). For further details, see the full paper.

Fluency is also an issue to be taken into account. In [69] the authors show that there appears to be a linear relationship between fluency measures and human judgements of proficiency. Also human-ratings of fluency have been found to be reliable with inter-rater correlation above 0.9 [70]. These results show the importance of measuring fluency as part of any pronunciation assessment exercise.

The current trend seems to be the use of the combination of different features. For example, in [71] a large feature set is used which includes duration, energy, pitch and pauses to detect word accents, and in a more recent work the same research team employs a discriminative approach that uses a large number of specialised rhythm features as well as general prosodic features to create a comprehensive metric of prosodic pronunciation quality [72]. In [73] the authors present a system that teaches fluency with several different methods of modifying the phoneme durations and f_0 contour of a learner's speech in order to demonstrate to the student what their pronunciation should sound like. Initial results from a pilot study seem promising. In [74] an SVM-based classifier is used for pitch accent recognition.

c) L1-dependency

A very interesting issue in CAPT research is the L1 dependency. Almost all the works in the literature are L1-dependent, since that approach has traditionally yielded a higher accuracy than L1-independent approaches. For example, in *AzAR* system [29] and *Euronounce* system [34] a predefined set of common pronunciation errors made by non-native speakers is considered, used to contrast L1-L2 phone confusion pairs. Also in [75] mispronunciation rules are manually set for a given L1/L2 pair, and they are used to cluster error rules using a decision tree.

L1-dependent research:

The use of L1 for training has two main advantages: Firstly, acoustic models that are a mixture of L1 and L2 can be used [46][76][77], which improves speech recognition accuracy and so enables recognition of less constrained utterances; this allows for greater freedom in the selection of pronunciation learning exercises. Secondly, the set of common pronunciation errors tend to be typical for a given L1 and very different between different L1, i.e. a Basque speaker will make very different English pronunciation errors than a native speaker of Amharic or Chinese. Thus, knowledge of L1 enables to provide tailored pronunciation exercises. For example, in [78] a tool called *L1-L2map* is described which contains manually entered data on likely mispronunciations for a given L1 when learning Norwegian. This data was then used to create a list of expected pronunciation errors. Likewise, in [79] a similar analysis is conducted to identify L1-specific groups of common errors for students of Dutch.

Different approaches have been described to automate the process of identifying typical error patterns for a given L1-L2 pair: in [80] and [81] [82] mispronunciation rules were automatically generated aligning canonical pronunciations with manually annotated pronunciations of non-native speech. Such rules are then used in extended recognition networks to identify pronunciation errors. This has the advantage that if an error is identified, the type of error is also known and can be used for diagnosis. In [83] an alternative method to generate mispronunciation lexica is explored. They use joint sequence multigrams to perform a 'grapheme to mispronunciation' conversion and showed that this approach can slightly improve performance both in terms of accuracy, as well as reduction in false alarm and false rejection rates. However, all these approaches still require a manually annotated corpus of non-native speech which is expensive and time-consuming to create.

Witt, in a study presented in the International Symposium on Automatic Detection of Errors in Pronunciation Training (IS-ADEPT) in 2012 [36], explains that there has been limited work that is L1-independent and yet has similar performance to methods based on knowledge of L2. Besides, the cost of non-native database collection and annotation is very high and does not scale. So the challenge is to develop methods that derive a set of likely errors for a given student knowing his native language, without requiring an annotated database for this L1/L2.

Witt concludes that one of the largest remaining challenges is the need to carry out research for a CAPT system that is L1-independent, or at least easily configurable for a different L1, without requiring a manually annotated non-native database. This idea fits in well with the philosophy of this thesis, because only a general-purpose ASR database is available to develop CAPT (and SGP) technology for Basque.

In the very first approach of our research in Germany [68], we defined a Basque curriculum, according to the main guidelines of *AzAR* (for German and Slavonic languages), including the most important phonetic and phonological aspects that a Basque student should cope with. However, the *AzAR* system was trained with non-native speech, and we did not have such a possibility for Basque. This led us to start research on L1-independent data.

L1-independent research:

Likelihood-based pronunciation scoring has the advantage of being L1-independent and very easy to compute. Many works employ classifiers for specific phoneme pair contrasts that represent common error types (see previous subsections). However, they have the drawback that common errors for a given L1-L2 pair have to be known, and that separate classifiers for each error type are necessary.

In [68] we introduced a simple method to make the CAPT system L1-independent. The GOP distributions of incorrectly pronounced phonemes are generated by introducing controlled changes in the system dictionary, i.e. replacing one phoneme in a particular position with another one of the same phonetic group (vowels, plosives, nasals, liquids and sibilants), at random. The underlying idea of this method is that a phoneme is incorrectly pronounced if it is acoustically closer to another phoneme of the same language. Thus, the distribution pairs of the GOP scores of each phoneme are obtained: correctly pronounced phoneme distribution and incorrectly pronounced phoneme distribution. The decision thresholds are obtained by calculating the equal error rate (EER) of both distributions (for further detail, see Chapter 6). This method is not as accurate as the L1-dependent ones, but it has the advantage that is not L1-dependent, and that no previous knowledge is needed about typical errors for a given L1-L2 pair.

2.2.2 Spoken Grammar Practice (SGP)

According to second language (L2) acquisition theories, naturalistic, implicit learning does not seem to be enough to achieve high-quality L2 proficiency in adults; explicit instructions are also necessary [84][85]. In the case of oral proficiency, to provide sufficient instruction and feedback is more time-consuming than in other skills, because it requires interaction with an individual tutor. This is one of the reasons why the implementation of SPG in CALL systems becomes so useful.

A very popular strategy from the initial systems consists in eliciting constrained output from learners by letting them read aloud an utterance from a limited set of answers presented on the screen or by allowing a limited amount of freedom in formulating responses. It is modelled by making the ASR decode through a finite state grammars in order to find the most probable path. This approach works well when the errors made by the student are completely predictable.

An early example of using finite state grammar was presented in [86]. The paper described a dialogue system aimed at teaching spoken Japanese, where speech recognition was used to analyse the student's answer at each stage of the dialogue. In [87], the *Subarashii* system is described, an experimental computer-based interactive spoken-language education tool. The system posed simple problems in written English which students had to solved by speaking an appropriate Japanese sentence. Since the set of situations was fixed, a finite state grammar was used to recognise correct and incorrect speech inputs for each situation. This system implemented ASR to enable communication between the computer and the student, but the student got no CF.

Another example of working with a limited search space was the CALL system named *Let's go* developed by the Carnegie Mellon University in 2004 for English [88]. This system included an algorithm to generate corrections that are as close as possible to the user utterance and provide CF by putting emphasis on the erroneous words. A list of target sentences was available beforehand. Besides, English acoustic models were also adapted with speech from non-native calls to the system, covering a variety of accents including speakers from Japan, India, Germany and China.

Another work in 2006 [89] also followed a two-step generation-based framework: given a possible ungrammatical input, the first step paraphrased the input into an over-generated word lattice, licensing possible corrections. The second step used language models and parsing to select the best rephrasing: a small set of N-best candidates were produced, which were then reranked by parsing using a stochastic context-free grammar.

Later in 2009, the CALL system for Japanese described in [90] aided students learning Japanese by creating their own sentences based on visual prompts. It was designed to detect lexical and grammatical (SGP) errors, and they receive CF about their mistakes. Questions were dynamically generated along with sentence patterns of the lesson point. A decision tree-based method was incorporated to predict possible errors made by non-native speakers, generating word grammars for the speech recogniser.

In [91], a new method is described to detect syntactic errors (in Dutch). The main idea was to generate an inventory of (syntactical) errors made by non-native speakers by analysing utterances from a corpus of non-native speech. The method made use of part-of-speech (POS) tags to label the words in each utterance, and an algorithm that matches words in two utterances: the (correct) target utterance and the (possibly erroneous) realisation of the utterance. This information was used to select errors and develop exercises for CALL systems.

The *DISCO* project (Development and Integration of Speech technology into COursEware for language learning) [92] aims at implementing SGP and CAPT in Dutch, and generating appropriate and detailed CF. Regarding SGP, syntax exercises are implemented making the student speak a group of words in a syntactically correct order. For these exercises, the speech recogniser determines which of the paths of a finite state grammar that includes all permutations of the word groups as paths is the most likely. In morphology exercises, a whole sentence is presented on the screen, but for one word a multiple choice list containing alternatives for that word is presented. Here, the ASR grammar includes alternative paths as well. In [93], the evaluation of a tool based in this system is described, where spoken practice of word order is offered (for Dutch grammar). The conclusion is that the system is successful in providing L2 speaking practice.

The first CALL system that implemented utterance verification (UV) was introduced in 2009 [94]. In this system, constrained responses were elicited from L2 learners. It used confidence scores along with a predefined list of possible responses for each exercise, to take the best ASR result from a predefined list of possible responses, and verify the correctness of the utterance in a second phase, in order to detect possible errors. The errors thus detected were shown to the student (CF). This system used acoustic likelihood ratio (LR) as confidence score [95]:

$$LR = \frac{p\,(x|u_1)}{p\,(x|u_{FPR})} \tag{2.1}$$

in which u_1 is the 1-Best decoding result given the signal x, and u_{FPR} is the optimal phone string found using free phone recognition (FPR). This predictor represents that when the input speech is not modelled as a path in the search space, the likelihood $p\,(x|u_1)$ is smaller relative to $p\,(x|u_{FPR})$ than when it is modelled. It estimates the posterior probability of the utterance given the speech signal x where $p\,(x|u_{FPR})$ is an estimation of the probability of x.

Current SGP systems do not provide real-time interaction. This means that the student has to utter a whole sentence before obtaining a CF. However, getting instant feedback at the same time as the student is solving exercises aloud could give the opportunity to the learner of making corrections on the fly, instead of waiting until a whole sentence has been uttered, sent to the server and feedback is received. That is the reason why a new UV-based technique has been introduced in this thesis: the Word-by-Word Sentence Verification (WWSV) [96]. This technique allows students to solve grammar exercises on the fly, since thus students can change their hypothesis about a sentence order (for example) in the same moment that an incorrect word is uttered and the system will give instant feedback about it. For further details, see section 4.3.

2.3 On-line implementation

The initial version of our CALL system was a software that had to be installed in a laptop (see Chapter 6). This has the advantage that no Internet connexion is required, but the process of updating and maintaining the software is harder compared with systems that work remotely. If signal processing occurs in a server, more powerful computers can be used remotely so that the processing time does not depend on the local processor. The current trend seems to be a web (client-server) configuration, downloading a web page from a HTTP server and establishing a websocket connection with a *Node.js* server to send binary audio data. This is explained in Chapter 7.

The web configuration involves some considerations:

- **On-line (real time) Voice Activity Detection (VAD)**: WWSV exercises have to be solved on the fly, so that the decision between which segment is speech and which is not has to be made instantly. In this thesis, a novel method has been devised: the Multi-Normalisation Scoring (MNS) method, which consists of generating multiple observation likelihood scores by normalising the MFCCs using means and variances computed from different speech datasets. The observation likelihood vectors thus obtained can characterise the behaviour of the speech and non-speech frames in different conditions. Further details about MNS-based VAD can be found in Chapter 8.
- **On-line (real time) cepstral normalisation**: the web implementation of an ASR-based CALL system involves that the audio signals that the students will send to the server will be recorded by different microphones. So some kind of normalisation must be applied in order to compensate for the acoustic differences (channels, background noise, etc.) of the incoming signals. The most common practice is to perform Cepstral Mean and Variance Normalisation (CMVN) of the extracted features, but an on-line version is also needed in order to be able to pass audio frames to the speech recogniser with no delay. A MNS-based CMVN has been proposed in this thesis, which yields encouraging results. For the details, see Chapter 9.

2.4 Summary

Many works in the literature use students' mispronounced data to adapt acoustic models or calculate thresholds between correctly and incorrectly pronounced phones (or words). However, the development of specific databases to create CAPT applications is a very expensive and time-consuming task, and requires a deep previous work on students' errors. Basque is regarded as an under-resourced language, and currently there is no suitable database for the development of speech-technologies for CAPT systems. The only available acoustic database recorded by microphone for ASR-related applications in Basque is the *Basque Speecon-like* database, recorded in an office environment and available only for research.

Furthermore, the trend in the design of current CAPT systems seems to be L1-independence. This is due to the fact that it is easy to cover some specific usual errors of the speakers of a particular language, but the need for a more global system remains.

Regarding SGP, there are applications specifically designed over ASR to practice aspects of L2 learning such as morphology and syntax. In some of them, ASR technology is used to detect students' lexical and grammatical errors and thus provide feedback; other applications use confidence scores along with a predefined list of possible responses for each exercise. However, there are few systems that provide real-time interaction at the same time as the students are solving exercises aloud.

In this thesis, we have devised a method to develop a tool to solve grammar exercises orally on the fly (Chapter 6), published in [96]. We have call it *Word-by-Word Sentence Verification* (WWSV), and it aims to verify a sentence uttered by a language learner in real time, word by word, in order to display the verified word as soon as it is detected. In this case, the feedback is the uttered word itself, and a correct response of the system is crucial.

Both the CAPT and the WWSV systems have been implemented in a remote web server, in order to see if they could be useful for Basque language students to improve their oral and grammar skills. This consequently involves overcoming new challenges, as the on-line (real time) VAD devised in this work, explained in Chapter 8 and published in [97], and the on-line (real time) CMVN, explained in Chapter 9.

PART II

The initial system

CHAPTER 3

The acoustic database and the phone inventory

3.1 The acoustic database: the *Basque Speecon-like* database

At the time of the presentation of this work, the only publicly available speech database for the development of speech recognition systems in Basque is the *Basque FDB-1060* database [98], which was designed according to the specifications of the European *SpeechDat* project [99] and it is distributed by *ELRA* through its repository[1]. The *Basque FDB-1060* database was recorded over the fixed telephone network, hence it does not meet the requirements of this thesis, since the acoustic models that could be obtained from it would not be optimal for microphone-input systems.

However, there are other acoustic data for Basque. In 2005, the Basque Government started a program, called *ADITU*, that specifically aimed at developing speech recognition and synthesis technologies for the Basque language. The *ADITU* program supported the creation of two new speech databases: the *Basque Speecon-like* database, recorded in an office environment according to *Speecon* specifications, and the *Basque SpeechDat MDB-600* database, recorded through mobile telephones according to the previously mentioned *SpeechDat* specifications. Both databases belong to the Basque Government and are not publicly available.

The *Basque Speecon-like* database was recorded in an office-environment, so it is suitable for the requirements of this thesis. It was created following the specifications of *Speecon* [100], a shared-cost project funded by the European Commission under Human Language Technologies, which was a part of the Information Society Technologies Programme (IST-1999-10003). The *Speecon* project was launched in 2000 and focused on collecting resources for training speech recognisers. Since the final goal of *Speecon* was the development of voice-driven interfaces for consumer applications, audio files were recorded via microphone at different distances. Under this framework, databases were produced for 20 European languages, among which Basque was not included. *ADITU* was aimed at creating a similar database to develop ASR systems for Basque.

[1] http://portal.elda.org/en/catalogues/

The next sections explain the main characteristics of the *Basque Speecon-like* database. For further details, please refer to [101].

3.1.1 Database contents

The *Basque Speecon-like* database consists of two main parts, one containing read speech and the other spontaneous speech. The read subset can be divided into two: core words, and phonetically rich sentences and words:

- **Read core words**: This set includes task-dependent words related to electronic devices, such as *aktibatu* (activate), *berrabiarazi* (restart), *aukerak* (options) etc. This category also comprises digit sequences, amounts, times and dates, natural numbers, spelling, two lists of frequently used street and city names, and a set of e-mail and frequent web addresses.
- **Read phonetically-rich sentences and words**: This set consists of phonetically balanced sentences and words, extracted from a text corpus created from newspapers, nowadays literature and oral transcriptions.
- **Spontaneous speech**: This set consists of around 5 minutes of spontaneous speech. It contains answers to some predefined questions or prompts about dates, time, spelling, companies and people names, cities, phone numbers, languages and yes/no answers, as well as a short spontaneous story about the speaker's personal interests and hobbies, films, TV series, etc. Some scenarios were also proposed related to bank transfers, hotel and travel booking, cinema tickets etc.

A breakdown of database contents is shown in Table 3.1.

Table 3.1: *Basque Speecon-like* database contents (number of utterances per speaker)

	#utts/spkr
Read speech	
General-purpose words and phrases	32
Application-specific words and phrases	212
Phonetically rich sentences	40
Phonetically rich words	8
Spontaneous speech	
Free spontaneous items	6
Elicited answers	17

The main difference between the read and the spontaneous subsets is that spontaneous speech is strongly dialectal for native speakers, whereas almost all read speech recordings involve standard Basque; an interesting fact for the development of CALL systems.

3.1.2 Recording platform

The *Speecon* specifications recommended to record speech signals in four different environments (office, public places, entertainment and car), to cover diverse application scenarios of ASR technology. Nonetheless, the Basque Government considered that the most interesting applications involved office environments, so recordings were made in closed rooms using a desktop computer as recording platform.

Additionally, the recording set-up was simplified, since only two microphone channels were recorded simultaneously: *close-talk* (by means of a head-mounted microphone) and *desktop* (by means of a microphone located at 1 m from the speaker), whereas the *Speecon* standard specifies four: *close-talk*, *lavalier*, *desktop* and *far-field*. A *Shure SM10A* was used as *close-talk* microphone and a *Shure SM58* as *desktop* microphone (with a *Shure FP11* microphone-to-line amplifier). Audio signals were acquired at 16 kHz and quantised using 16-bit PCM coding.

3.1.3 Database size

As far as the amount of data is concerned, the *Basque Speecon-like* database contains 23.8 GB. The documentation files, which comprise the transcription files and some design information of the database, are approximately 20 MB. The remaining data is divided in two equal blocks: the files corresponding to the *close-talk* microphone channel (short distance), and the ones corresponding to the *desktop* microphone channel (medium distance). The size of each block is 11.8 GB.

Regarding the duration of the files of the database, the whole database contains 109.95 h of recordings, almost 30 min per speaker on average. Speech comprises 52.67 h out of the total duration, and non-speech (silence, breaths and the like) 57.28 h. Regarding the speech, 30.37 h correspond to the read speech subset, and 22.30 h to the spontaneous speech subset.

The spontaneous speech content is 42.34 % of the whole speech content of the database. This means that a significant amount of the speech in the database includes many and diverse disfluencies. Table 3.2 summarised the amount of hours of each part in the database.

Table 3.2: *Basque Speecon-like* database contents (hours)

		h	(total h)
Speech	Read	30.37	52.67
	Spontaneous	22.30	
Non-speech	Silence	47.65	57.28
	Breaths, mic. sounds etc.	9.63	

3.1.4 Distribution of speakers with regard to dialect region and competence level

The distribution of the speakers in the *Basque Speecon-like* database is an issue that must be handled with care, for them to be representative of the community of potential users. The Basque language is very complex with regard to demographics, since it presents many dialectal variations and significant variabilities even inside the same dialect. The standard Basque is relatively new, to the point that native Basque speakers may not be used to speak standard Basque. That is why most native speakers, though capable of using standard Basque in elicited recordings, use their own dialectal variety in spontaneous speech recordings.

The distribution of speakers with regard to dialectal region and competence level is shown in Table 3.3. For historical reasons, there is an important amount of non-native Basque speakers with diverse competence levels. This fact also contributes to the disfluencies found in the spontaneous speech part of the database. The documentation of the database also includes a rough binary classification between high and low level non-native speakers, which is also shown in Table 3.3.

Table 3.3: Distribution of speakers with regard to dialect and competence level in the *Basque Speecon-like* database.

	Native	High level non-native	Low level non-native	Total
Gipuzkoa	85	13	2	100
Bizkaia	49	32	15	96
Nafarroa	14	6	3	23
Araba	0	3	4	7
Others	1	2	1	4
Total	149	56	25	230

Due to recruiting issues and schedule requirements, the final distribution of speakers did not exactly match the design. As a result, some dialect regions (Nafarroa and Gipuzkoa) were over-represented, whereas others (especially Araba) were under-represented. In any case, the resulting distribution is still useful and quite representative of the demographics of Basque.

It is worth mentioning that Northern dialects were not considered in the design of the *Basque Speecon-like* database. There are three dialects in the administrative area of France, which are referred to as Northern dialects of Basque, and no one was included in the database. No need to say that a proper Basque acoustic database should also include speech from people from that area.

3.1.5 Distribution of speakers with regard to age and gender

Although in the *Speecon* specifications both adult and children speech were considered, only adult speakers were recorded in the *Basque Speecon-like* database. The total amount of speakers is 230 (127 female + 103 male). Table 3.4 summarises the gender distribution across the age groups defined in the standard.

Table 3.4: Distribution of speakers with regard to age and gender in the *Basque Speecon-like* database.

	Female	Male	Total	%
15-30	67	38	105	45,65
31-45	48	51	99	43,04
46+	11	14	25	10,87
Unknown	1	0	1	0,44
Total	127	103	230	100
%	55,22	44,78	100	

As can be seen in Table 3.4, the *Basque Speecon-like* database shows a slight imbalance with regard to the relative amounts of female and male speakers. However, the deviation is very small, so we assume that it will not entail any negative effect on the acoustic models created from the database.

3.1.6 Annotation

The *Basque Speecon-like* database provides all the recordings but only part of the transcription files. So part of the annotations have had to be done at the very beginning of this thesis, including an improved lexicon and improved transcriptions of the spontaneous speech recordings.

Orthographic Annotation

In the case of read speech, the orthographic transcriptions of the audio files were available beforehand. Thus, these transcriptions were just checked and corrected when needed. In the case of spontaneous speech, the whole transcriptions were manually created from scratch and later checked to fix errors and to increase consistency.

Acoustic events

The orthographic transcriptions include acoustic events and word deformations (see Table 3.5). This allows to keep as much speech in the database as possible, avoiding the need for taking out recordings from the database.

Table 3.5: Labels used to mark acoustic events and word deformations in the Basque Speecon-like database.

	Symbol	Meaning
Speech events	{FIL}	Filled pause
	{FRA}	Word fragment
	{LNT}	Lengthened word
	{TRC}	Truncated word
	{UNI}	Unintelligible word
Non-speech events	{BRE}	Breath (and laughs)
	{INT}	Intermittent noise
	{SPK}	Speaker noise (lips smacks etc.)
	{STA}	Stationary background noise

Phonetic Transcription

An improved lexicon has had to be created containing all the orthographic entries together with their broad phonetic transcriptions. The improved lexicon has been automatically generated using the Aholab grapheme-to-phoneme (G2P) transcriber for Basque, including as alternatives the different pronunciations that the same standard word shows in different dialects, in order to cope with the dialectal variations appearing in the spontaneous speech part of the database.

The lexicon currently contains 29 626 different lexical entries (in standard Basque), with 122 542 different pronunciations. This means that each word has, on average, 4,14 different pronunciations, which account for dialectal variations, revealing the complexity of Basque from the point of view of its use in the daily life.

The *Basque Speecon-like* database considers a basic set of 36 different phones. 35 of them correspond to the basic set of SAMPA Basque phonetic alphabet[1] [102]: *p*, *b*, *t*, *c*, *d*, *k*, *g*, *tS*, *ts*, *ts‘*, *gj*, *jj*, *f*, *B*, *T*, *D*, *s*, *s‘*, *S*, *x*, *G*, *m*, *n*, *J*, *l*, *L*, *r*, *rr*, *j*, *w*, *i*, *e*, *a*, *o*, *u*. One last phone is also considered from SAMPA Basque alphabet: the *Z* allophone, which belongs to the western dialects.

3.2 The phone inventory

The phone set included in the *Basque Speecon-like* database has been simplified for this work. Out of the initial 36 phones a final set of 30 phones has been considered. This section explains the reasons and characteristics of the chosen phones. The phonetic symbols used correspond to SAMPA Basque phonetic alphabet.

[1] http://aholab.ehu.eus/sampa_basque.htm

3.2.1 Some considerations

- **Voiced plosive vs. approximant**: The realisation of voiced plosive and approximant phonemes depend on the context of their location. The approximants B, D and G occur in specific contextual locations, as for example between vowels, and b, d and g in the rest of the cases. Henceforth, they can be considered as allophones of the same phoneme. Since context-dependent phones will be used, a single phone has been used to represent each pair of allophones.

- **Semi-vowels**: Semi-vowels are sounds that are phonetically similar to vowel sounds but function as syllable boundary rather than as the nucleus of a syllable, as is the case in diphthongs. The only existing diphthongs in Basque are those finishing in close vowels (i eta u), and are falling diphthongs which start with a vowel and end in a semi-vowel. Consequently, there are two semi-vowels in Basque: j and w, which correspond to the vowels i and u, respectively. A vocalic pair ending in i or u is only an hiatus when an h is between them. Nevertheless, the h, although formerly pronounced in Basque, nowadays is only pronounced in the northern dialects. This implies that the words containing hiatus are in most cases diphthongised, as in the word *ehiza* (hunting), where the canonical transcription is /*eis‘a*/ (three syllables), but the usual realisation is /*ejs‘a*/ (two syllables). As a result, very few hiatus realisations have been found in the database. So a single phone (i and u) has been used to represent both the vowel and the semi-vowel pair, since, in view of the above, they only depend on the context.

- **The allophone** Z: This allophone, the only voiced sibilant in the Basque phoneme inventory, appears only in western dialects, in a very specific context: when a word finishes with i and the article '-a' is added to the word, the Z is inserted between them. This does not happen in all the western dialects, and there are very few occurrences in the database. Consequently, it has been removed from the final phone inventory.

- **The palatalisation of the phonemes** n **and** l: A phonological process occurs in the southern dialects, which is also accepted for the standard Basque: the phonemes n and l palatalise when they have a preceding i. This phenomenon occurs always, except in a few borrowed words and proper names. Anyway, the distinction between the non-palatal and the palatal versions of these phonemes has been deemed interesting to keep, in an attempt to obtain distinct acoustic models and detect what beforehand can be predicted that will be a very frequent error in the pronunciation evaluation task.

- **The case of '*j*'**: This grapheme is differently uttered in Basque, but the standard gives preference to the voiced fricative jj, which is dominant in the eastern varieties. Nevertheless, the velar fricative x of the central dialect is widely used, and so both allophones have been kept in order to obtain distinct acoustic models for their use in the CAPT task.

3.2.2 Final phone inventory

Account taken of all the above, the final phonetic set chosen for this thesis is the list shown in Table 3.6, grouped by main features. See SAMPA Basque phonetic alphabet for further detail.

Table 3.6: Final phone inventory to develop acoustic models from the *Basque Speecon-like* database.

Group	Phone
Vowels	*a*, *e*, *i*, *o*, *u*
Unvoiced plosive	*c*, *p*, *t*, *k*
Lateral	*l*, *r*, *rr*
Affricate	*ts‘*, *ts*, *tS*
Nasal	*m*, *n*, *J*
Palatal	*L*, *jj*, *gj*
Voiced plosive	*b*, *d*, *g*
Fricative	*f*, *x*, *T*, *s‘*, *s*, *S*

3.2.3 Final acoustic event list

With regard to the acoustic events, only two speech events have been selected from the documentation of the *Basque Speecon-like* database: the {FIL} and {UNI} events. The words containing one of the {FRA}, {LNT} or {TRC} labels have been replaced by the {UNI} label, which will serve to collect all confusing speech segments in the process of creating the acoustic models. Thus, the {UNI} label will not be used in posterior recognition experiments, it will only be a kind of garbage collector for the training process.

Table 3.7: Final acoustic event inventory to develop acoustic models from the *Basque Speecon-like* database.

	Symbol	Meaning
Speech events	{FIL}	Filled pause
	{UNI}	Unintelligible word
Non-speech events	{BRE}	Breath (and laughs)
	{MIC}	Microphone taps and touches
	{SPK}	Speaker noise (lips smacks etc.)
	{STA}	Stationary background noise

In respect of non-speech events, four labels have been used: the {BRE}, {SPK} and {STA} labels from the *Basque Speecon-like* database, and a new one: the {MIC} label, which will account for microphone taps, touches etc. The decision of creating this new label comes from the idea of using the close-distance microphones signals for this thesis, both to create acoustic models and to use the final system.

Table 3.7 shows the list of the selected acoustic events.

3.3 Conclusions

The *Basque Speecon-like* database is a general-purpose ASR database for Basque, but it is the one chosen for this thesis, since it is the only database publicly available. The database has two parts: a *read* part and a *spontaneous* part. Although the *read* part contains some variations in the pronunciation of certain phonemes, the spontaneous part is strongly dialectal and contains a big amount of phonetic variations. In ASR, phonetic variations can be modelled using the same acoustic model. However, for CALL systems, "clean" acoustic models are needed, in consequence some work had to be done on the database.

The database has the advantage that it contains speakers labelled depending on their language skills. There are three labels for each speaker: native, high-level (non-native) and low-level (non-native). This information is very useful to create acoustic models for CAPT choosing only native speakers, or to test the CAPT system using speakers with different Basque levels.

CHAPTER 4

The underlying ASR system: *AhoSR*

4.1 Introduction

In this chapter, the underlying speech recognition technology of the CAPT and SGP systems is described: the *AhoSR* speech recognition system. *AhoSR* is a speech recognition decoder developed from 2010 onwards in the *Aholab* research group, which aims at providing a flexible computing environment for ASR-based applications and research. Basically, *AhoSR* is a modular speech recognition decoder written in *C++*, which is based on Hidden Markov Models (HMM) and uses Mel-Frequency Cepstral Coefficients (MFCC) as acoustic features. It is designed to cope with different tasks, like phonetic recognition, word-grammar based recognition, or large vocabulary continuous speech recognition, where the language model information is decoupled and incorporated at run-time. Utterance verification techniques can also be applied on the basic tasks, mainly devised for CAPT and SPG applications. The decoding process is performed frame-synchronously (breadth-first search, BFS) by means of the token-passing paradigm [103] and the one-pass beam-search strategy [104]. *AhoSR* is a multi-platform system which can be used in Unix-like and Microsoft Windows systems, and accepts both direct audio (through an audio recording device and via socket connection) and *wav* files as input. Since 2014 there is a stable version of *AhoSR* which has been presented in [105]. For more information, such as evaluation experiments or applications, please see that paper.

Several open-source tool-kits are nowadays available for researchers working on the field of ASR, among them *HTK* [106], *Julius* [107], *Kaldi* [108], *RWTH ASR* [109] and *Sphinx-4* [110]. However, there are some important issues that led us to develop our own system. The main reason was the fact that all of the mentioned tool-kits need to be tuned when a specific use in a non-classical ASR application is required. For example, ASR-based CAPT and SPG applications make use of verification scores which can not be easily obtained with such a tool-kit, and commonly a parallel graph is built in order to compute these scores (see section 4.3). In many cases, the effort to tune the existing tool-kits can be as important as building them from scratch.

On the other hand, the tool-kits mentioned above are optimised to work with word-based language models, typically N-grams, which are useful for languages with no overt inflection –e.g. Chinese–, with minimal inflection —e.g. English—, or not highly inflected —e.g. Spanish—. However, in highly inflective or agglutinative languages (like Basque, among many others), words are built concatenating several prefixes and/or suffixes to the word roots, leading to millions of different but still frequent word forms [111]. Furthermore, the need of using such big vocabulary sizes causes a high amount of OOV (out-of-vocabulary) words, which have direct impact on the recogniser efficiency [112]. Different approaches are being tested, mostly based on using sub-word units as basic speech recognition units, as explained in [113] for Turkish, in [114] for Arabic, in [115] for Hungarian, or in [116] for Tamil, or in [117] for Basque. The introduction of such a sub-word units based language models requires the modification of the search space and fine control of the propagating paths.

There is also the issue of availability for commercial uses, which varies from one tool-kit to another. If an ASR-based CAPT or SPG tool is to be integrated in an existing language learning tool, or a migration to an embedded system is foreseen, full availability of the source code is crucial. Taking all these aspects into account, we considered it very convenient to develop an adaptable recognition system where different approaches could be applied and tested, and which would keep all the doors open for future developments.

In this chapter, firstly the overall system architecture of *AhoSR* is depicted, and each block is described in detail. Then, the modifications made for the Word-by-Word Sentence Verification (WWSV) task are explained: the addition of a parallel graph for verification, the improvement and adaptation of the search graph for sentences, and the decision making procedure in the on-the-fly verification process.

4.2 System architecture

The overall architecture of *AhoSR* is modular, so that modifications and adaptations can be easily applied in each block separately without affecting the rest of the modules. There are four primary modules in *AhoSR*: the *Main Manager*, the *Front-End*, the *Linguistic Knowledge Base*, and the *Decoder*. The *Main Manager* sets and manages the values of different parameters of the recognition process, and controls the sequence of execution of the different parts of the recogniser. It processes the data received from the *Decoder* and yields a final result. The *Front-End* parameterises the input audio signal (both direct audio and *wav* files) into a sequence of features and passes them to the *Decoder*. The *Linguistic Knowledge Base* stores three types of data: the acoustic models, the lexicon, and the language model, which are used to create the search graph. The *Decoder* firstly creates the appropriate search graph for the task and then it takes the feature vectors from the *Front-End* and begins and controls the actual decoding process over the search graph. It sends the decoding details and results to the *Main Manager*. The overall block diagram of *AhoSR* is shown in Figure 4.1.

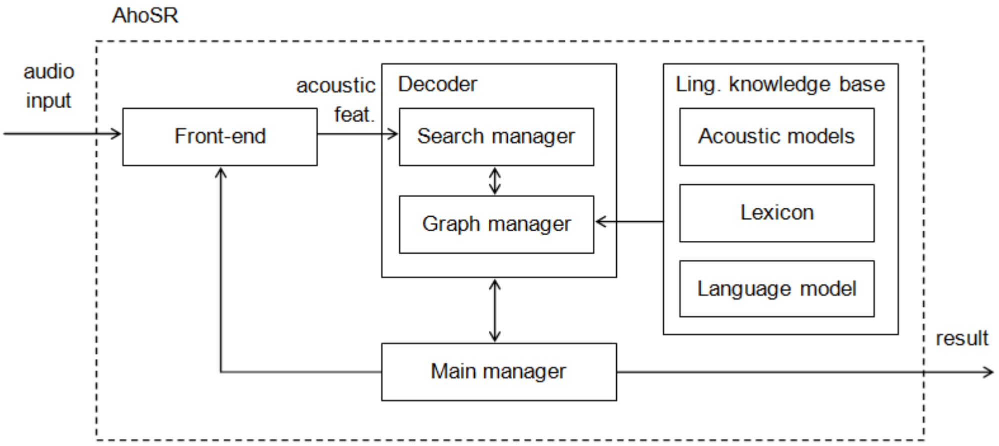


Figure 4.1: System architecture of *AhoSR*. The main blocks are the *Main Manager*, the *Front-End*, the *Linguistic Knowledge Base*, and the *Decoder*. The communication between these modules is depicted.

4.2.1 Main Manager

The *Main Manager* is responsible for managing the communication between different parts of the recogniser. Firstly, it reads a configuration file where the user sets the values of different parameters, and it checks for potential incompatibilities among them. According to these values, the *Main Manager* configures and initialises each module, and decides which execution sequence the decoding process must follow. Then it initiates the decoding process.

During the decoding process, the *Main Manager* controls the communication between the different parts of the recogniser, and when the decoding process is finished, it processes the data received from the *Decoder* and yields a final result. The results can be shown or stored in different formats in accordance with the requirements of the user: with or without duration and score information, at phone-level or word-level, with or without verification scores, etcetera.

The most important configurable parameters are shown in appendix A.

4.2.2 Front-End

The purpose of the *Front-End* is to manage the feature extraction process, converting the input audio data into standard MFCC vectors with optional delta and delta-delta (also known as differential and acceleration) coefficients so as not to lose information in the dynamics [118]. Several parameter values can be set in order to obtain different MFCCs: the frame rate, the frame length, the number of mel bins, the minimum and maximum frequency cut-offs, etc. The generation of features and the management of the stack they are stored in are controlled by the *Main Manager*, which sends a message to the *Decoder* when new data are available.

The *Front-End* also supports other helpful techniques like different implementations of Voice Activity Detection (VAD) (see Chapter 8), and Cepstral Mean and Variance Normalisation (CMVN) for noise robust performance (see Chapter 9).

4.2.3 Linguistic Knowledge Base

In *AhoSR*, three different knowledge sources are used to create the search graph of the recognition decoding process:

Acoustic Models

HMMs [119] are used to model the acoustic sequential structure of speech signals, with local spectral variability modelled using mixtures of Gaussian densities (continuous density HMMs).

The parameters of a HMM are of two types: transition probabilities and emission probabilities (also known as output probabilities). The transition probabilities control the way the state j at time t is chosen given the state k at time $t-1$: $a_{kj} = p(s_j|s_k)$. The emission probability of producing observation x when being in state s_j at each time frame is: $b_j(x) = p(x|s_j)$, and is obtained calculating the probabilities of each mixture component corresponding to the same feature of the input speech vector. The topology of a three emitting state generic HMM is shown in Figure 4.2.

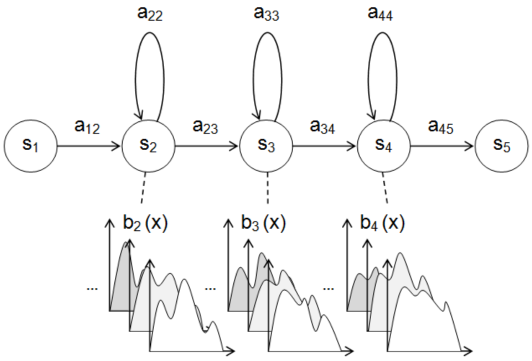


Figure 4.2: *AhoSR* manages continuous density HMMs as acoustic models. In the figure, three emitting state HMM topology, with transition probabilities a_{ij} and output pdfs $b_j(x)$.

AhoSR can manage HMMs modelling different types of word or sub-word units. Nevertheless, it is optimised for triphones, which account for the left and right context of a phone. One problem with triphone HMMs is that there is usually a large number of models for the amount of training data available. Furthermore, many triphone contexts are very similar. Regarding this, *AhoSR* allows the use of tied-state (also known as *senone*) HMMs, which are created by clustering similar states together [120]. Apart from taking the advantage of gathering more training material to robustly estimate each set of state output distribution parameters, the processing time reduces significantly.

The format of the acoustic models is HTK-compatible. This means that HMMs generated with the HTK tool-kit can be used in *AhoSR*.

Lexicon

The lexicon is a file that contains the mapping between the written representation and the pronunciation of a word. The pronunciations must be depicted as a sequence of units (words, syllables, phones etc.), all of which must match a given HMM. As a result, each word in the lexicon is described as a sequence of HMM states (see Figure 4.3).

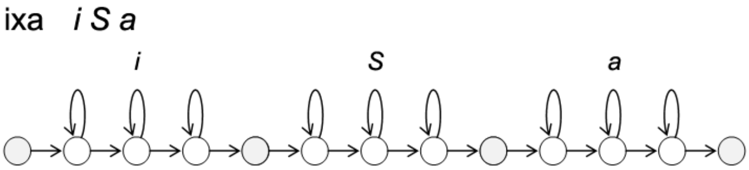


Figure 4.3: The Basque word **ixa** (*ex*, letter "x") is described in the lexicon like the HMM sequence /*i S a*/ (SAMPA Basque notation).

AhoSR also manages alternative pronunciations of words beyond the canonical representation, thus taking into account pronunciation variations due to, for instance, dialectal variations. The alternative pronunciations must be included in the lexicon as different entries with the same lexical word (and the corresponding pronunciation).

Language Model

Two types of language models are managed: context-free grammars and N-gram language models. The standard adopted for context-free grammars in *AhoSR* is the Augmented BNF (Backus-Naur Form) notation [121], which defines a syntax for representing grammars to use in speech recognition. On the other hand, ARPA format N-gram back-off language models can also be used to implement a statistical language model [122], which can be generated, for instance, with the SRILM tool [123].

The modular nature of *AhoSR* allows new grammar formats to be easily added to the system, without deep knowledge of the internal representation of the search space. This provides the possibility of easily testing new solutions for different tasks or for researching on agglutinative languages like Basque.

4.2.4 Decoder

The *Decoder* is composed of two primary modules: the *Graph Manager*, which controls the construction of the specific search graph for the required task; and the *Search Manager*, which picks up the incoming acoustic features from the *Front-End* and manages the decoding process over the search graph created by the *Graph Manager*. The results obtained by the *Search Manager* are sent to the *Main Manager*.

Graph Manager

The primary function of the *Graph Manager* is to create a search graph suitable for the task. Firstly, it translates the information of the *Linguistic Knowledge Base* into an internal data structure. Then, it creates the search graph by means of the information of the *Linguistic Knowledge Base*: using the language model information, the *Graph Manager* creates a suitable word-level net, composed of nodes and arcs, where each node represents a word and each arch the relationship between nodes. Then, each node of the word-level net is substituted by the corresponding HMM sequence representation given by the lexicon (alternative pronunciations are allowed). Finally, each HMM representation is linked with its corresponding HMM, thus obtaining a state-level net or final search graph comprised of nodes and arcs. For phonetic recognition, a special way of creating the search graph is used, where HMMs are considered as if they were representing a word.

The search graph can be compressed in order to obtain a significant reduction in the acoustic search effort. *AhoSR* allows prefix-suffix tree compression to be performed over the search space [124][125]. In this search space topology, the language model is a separate module which is consulted at run-time by the *Search Manager*. This characteristic makes *AhoSR* not only memory efficient, but also flexible in use.

In case utterance verification is needed as well, the *Graph Manager* is the responsible for creating a parallel search graph, through which verification scores will be obtained.

Search Manager

The *Search Manager* uses the token-passing algorithm [103] for the decoding process. It expands tokens through the search graph, making use of the standard Viterbi algorithm. Each token contains information about the search and provides a complete history of all active paths in the search. Besides, each token stores the overall acoustic and language scores of the path at a given point. During the search, each incoming feature frame is scored against the acoustic models associated with each token state, and low scoring branches are pruned. Two types of pruning have been implemented, which can also be combined: global beam pruning, which retains only paths with a likelihood score close to the best partial path hypothesis; and histogram pruning, which limits the number of active paths at each time frame by retaining only a predefined number of best paths [126]. A configurable number of tokens propagates in each state node or in each auxiliary node of the graph so that, for instance, an N-Best list can be obtained.

When choosing the verification performance mode, the *Search Manager* also computes GOP (Goodness of Pronunciation) scores, which are calculated as the duration normalised log of the posterior probability of a phone or sequence of phones over the acoustic segment [127]. To compute the GOP, two groups of scores are used: on the one hand the ones obtained in the main search graph during the recognition process, and on the other hand those obtained in a secondary graph, which consists of a free phone loop (see section 4.3).

4.3 Adaptations for the WWSV task

As explained in the introduction, *AhoSR* has to be tuned to implement the on-the-fly WWSV task. Firstly, a parallel graph has been added to the main graph in order to compute the verification scores. Secondly, the search graph to verify sentences has been improved, taking into account the co-articulation between words and the different pronunciations each word can have. Finally, the *Search Manager* has been tuned, in order to check when a word reaches a threshold value, and take the decision whether the word can be considered as verified or not, thus starting the same process with the next word in the sentence. This section explains each part in greater detail.

4.3.1 The parallel graph

As explained in section 6.2, the basic GOP measure for a phone q_i can be computed as in equation (4.1).

$$GOP\left(q_i\right) \approx \frac{1}{T_i} log \left[\frac{p\left(O_i|q_i\right)}{p\left(O_i|q_{j_{max}}\right)} \right] \tag{4.1}$$

where j_{max} is the phone model index that gives the highest likelihood for the segment to be evaluated. From this approach, this value is calculated using an unconstrained loop running in parallel with the main search graph, which can be composed of all the triphone models used in the decoding process. However, this would create a substantial delay in the system due to the large amount of triphone models. Consequently, a monophone loop is usually used, since they amount to only 30 models. Thus, the parallel loop is lightweight and does not create any delay.

4.3.2 Specific search graph

In the WWSV task, the user must think and build a sentence in a given time. Thus, the search graph in the decoding process must take into account three different situations when passing from one word to the next:

- **Silence between words**: The user pauses and a silence is inserted between words.
- **Juxtaposed words without co-articulation**: The user takes a very short pause between words which can not be considered as silence, but co-articulation does not occur between them.
- **Juxtaposed words with co-articulation**: Juxtaposed words are uttered without any break, and co-articulation occurs between words.

An example of the search graph designed to account for these three situations is depicted in Figure 4.4. The figure shows a three-word sentence ("asteartea, osteguna, larunbata": *Tuesday, Thursday, Saturday*) that takes also into account different pronunciations for each word.

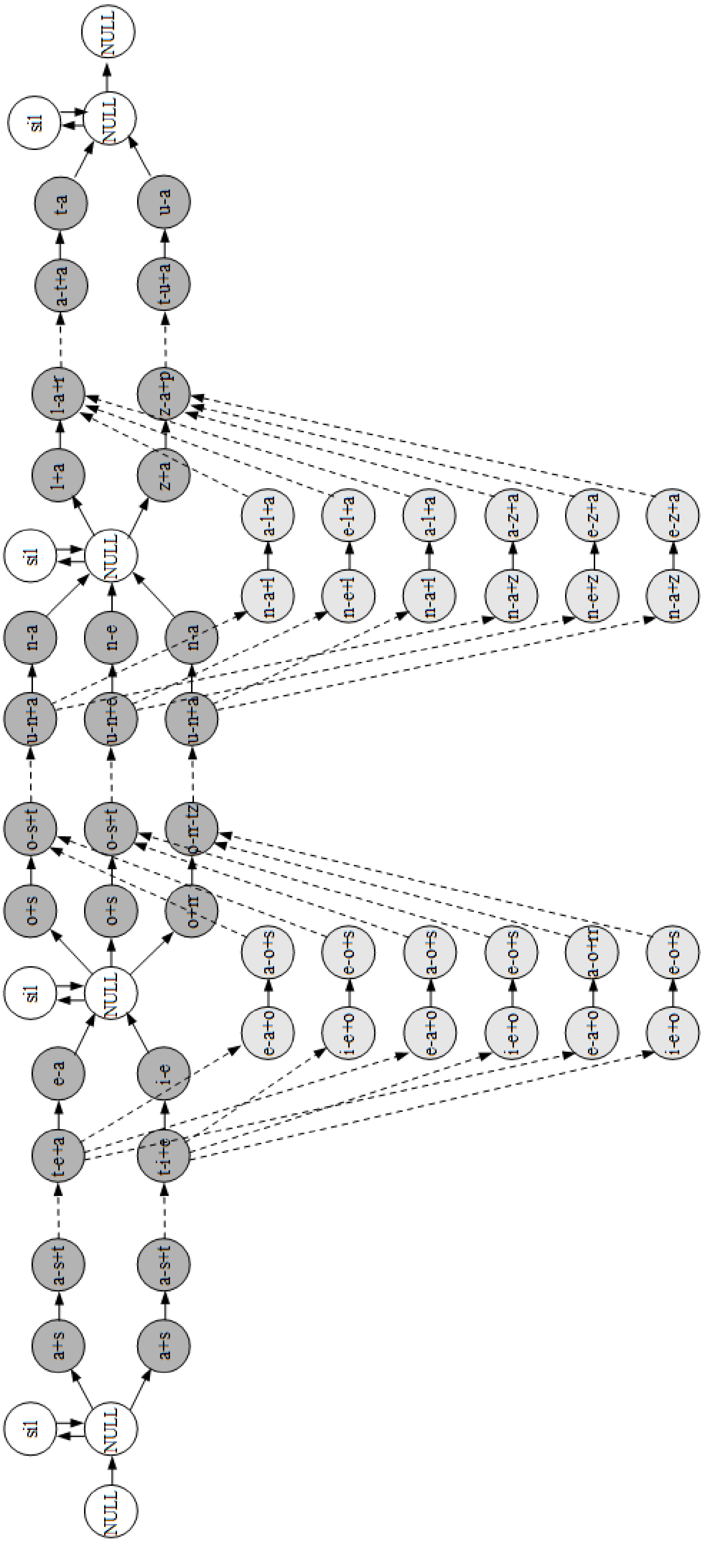


Figure 4.4: The decoding-network resulting of the sentence "asteartea, osteguna, ostirala".

In a realistic environment the student will make mistakes. That means that the verification system must be able to manage these extra voice segments in order to absorb the effects on the Viterbi decoder. Since the system will receive more voice frames than it expects, this must be modelled somehow. So an optional phoneme loop has been added to the main search graph of *AhoSR* at the beginning, at the end and between words, in the way shown in Figure 4.5. If this phone loop was not added, the segmentation resulting from the Viterbi algorithm would not be predictable in case there were user mistakes, and the verification or scoring could not be calculated over that segmentation.

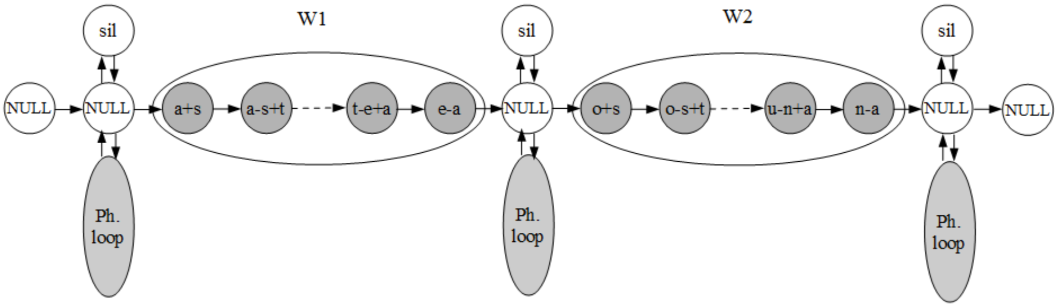


Figure 4.5: Free phone loops added in parallel to silence nodes in the search graph of *AhoSR* for the WWSV task.

4.3.3 Changes in the *Search Manager*

The *Search Manager* has to decide when the word that is being verified is actually that word. To do that, the system assesses the confidence scores at word level. When a maximum above a predefined threshold is detected in the confidence score curve, it performs a deeper verification at phone-level, using the segmentation given by the Viterbi decoder. If all the phones reach their corresponding threshold, the word is considered as verified, and it is immediately displayed. Then, the resources of the search graph related to the verification process are updated, and the processing of the next word begins.

In the event that any of the phones does not reach the threshold value, the current frame is ruled out, and the verification process continues with the next frame. The WWSV process is finished when the last word is positively verified.

4.3.4 The socket audio input

The audio input can be a *wav* file as well as a direct audio stream coming from an audio recording device or through a socket connection. It is worth mentioning that the socket input functionality has been added specifically for this thesis, in order to implement real time performing when accessing *AhoSR* through the internet. This functionality has given us the opportunity to install *AhoSR* in a server, and in consequence, it has been possible to develop some publicly available demonstrators, not only for WWSV based SPG tasks, but also for recognition tasks. They can be tested here: http://aholab.ehu.eus/users/igor/demos.html.

4.4 Conclusions

This chapter shows the structure and functionalities of *AhoSR*, the underlying speech recognition software used in this thesis, created and developed by the *Aholab* research group. Currently, it is based on HMMs and uses MFCCs as acoustic features. It is designed to cope with different task, like phonetic recognition, word-grammar based recognition, or large vocabulary continuous speech recognition.

For the purposes of this thesis, utterance verification techniques were also implemented to be run together with those basic tasks. To do so, a parallel graph has been implemented to obtain the GOP scores. For CAPT and WWSV tasks, specific search graphs have been added, in order to adapt to the needs of each of them.

In addition, sockets have been implemented in the audio-input module of *AhoSR*. This allows real time performing when accessing the recogniser through the internet, and so it gives us the opportunity for *AhoSR* to be installed on a server, with universal access.

All the details about *AhoSR* can be found in [105].

CHAPTER 5

The acoustic models: HMMs

5.1 Introduction

Acoustic models are the core of both the Word-by-Word Sentence Verification (WWSV) task and the pronunciation scoring task. In order to obtain GOP scores, both make use of a main search graph and an additional verification graph, which are built concatenating Hidden Markov Models (HMMs). Therefore, the final results will directly depend on the quality of the HMMs (basically, observation probabilities and transition probabilities).

In order to obtain accurate acoustic models, a properly labelled corpus is needed. As explained in section 3.1, the only public acoustic database recorded through microphone for Basque is the *Basque Speecon-like* database [101]. This database contains 230 sessions, one for each speaker, and it has been divided into two main blocks to carry out the experiments of this work: the *train* block which includes the first 155 sessions (74.10 h), and the *test* block, consisting of the remaining 75 sessions (35.85 h). The *train* block has been used to train both the HMMs and the GOP scores of section 6, and the *test* block to evaluate them.

As the *spontaneous speech* part of the database contains dialectal speech and it forms the 42.34 % of the whole speech content (see Table 3.2), alternative pronunciations have been created to cope with such variations. A number of earlier studies explored the use of alternative pronunciations in speech recognition experiments. On the one hand, alternative pronunciations, obtained typically by manual addition or by maximum likelihood learning, increase the coverage of pronunciation variability. On the other hand, they may also lead to greater confusability between different lexical items. These two opposing factors usually result in either no recognition performance improvement or a minor one compared with the use of standard dictionaries (e.g., [128]). However, the goal of this work is not the creation of good models for recognition, but for pronunciation scoring. So, this approach could be appropriate or, at least, a preliminary research work, to train good-quality discriminative acoustic models.

The quality of the acoustic models will be assessed using the Phone Error Rate (PER) score. Our goal is to reach the state-of-the-art values of this score, published in scientific papers for other languages. Doing a little background research on phonetic recognition, a paper published in 2011 shows the evolution of the results obtained in experiments on the TIMIT database from 1990 to 2011 [129]. It describes that the performance of the phonetic recognition results improved about 13% from 1990 to 2010, and that happened mainly in the first 5 years of research: in 1990 a minimum PER of 26.20 % was obtained, using discrete monophone HMMs [130], and near 1995 PERs of 22.50 % were achieved using continuous triphone-HMMs [131] and recurrent nets [132]. Since then until 2011, the improvement was very slight. The conclusions of that paper explained that it appeared that a bound of about 20 % would be hard to beat, which reflects the thought of the time. However, the new arrival of Artificial Neural Networks (ANNs) that year made it clear that considerable improvements were to happen. In 2015 a PER of 17.10 % was reported in [133] using Convolutional Neural Networks (CNN), and later that year a PER of 16.50 %, by the same author, using CNNs as well, but with some refinements [134].

This chapter is organised as follows: firstly, different ways of training HMMs are described, using different parts of the database, even at different stages of the HMM training process. The PER scores obtained by each HMM set are also displayed in this section, as well as a breakdown for every phone. The next section shows how feature-normalised models can be obtained in order to neutralise the effects of channel mismatching between the audio used for training the models and the input audio of the final user. To see the effects of the channel mismatching, an experiment is introduced in the next section, which consists in testing audio files recorded through a different microphone and distance. Finally, some concluding remarks are discussed.

5.2 HMM training

The phonetic unit selected to be modelled has been the widely used *triphone*. Triphones are context-dependent phones that are characterised by the influence that anterior and posterior phones generate on them. While in the word *Aholab*, which is pronounced as /*a o l a b*/[1], the phoneme /*a*/ of the first and the last syllable would be the same unit using monophones, it is differently modelled using triphones, since the context of each phone instance is different. The three part nature of triphones makes them ideal to be modelled with three state HMM. The first state of the HMM is used to model the left context of the phone (the right half of the left phone transition); the second or central state models the phone core; and the third state models the right context (the left half of the right phone transition). Figure 5.1 shows a graphical explanation of this.

The triphone HMMs have been created using the *HTK tool-kit* [106] with the feature vectors extracted from the signals by means of *AhoSR* [105].

[1] http://aholab.ehu.eus/sampa_basque.htm

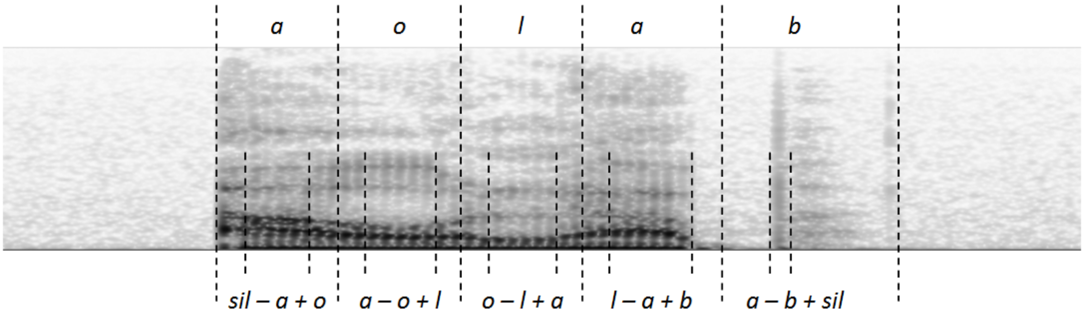


Figure 5.1: Spectrogram of a realisation of the word *Aholab* and its division into monophones (top) and triphones (bottom).

At the time of training the HMMs, two different subsets have been taken into account (see database contents in section 3.1.1):

- Subset R: The *read* subset of the database, consisting of items 25-316 in each session.
- Subset W: The whole *train* block (items 1-316 in each session).

The reason why this division has been considered is that, as explained previously, the *non-read* subset of the database contains a large amount of variations, and separate models have been trained in order to evaluate whether this subset is worth including or, otherwise, adds noise.

Additionally, for the training process of the HMMs, two dictionaries were automatically generated from the *train* block of the database using the Aholab grapheme-to-phoneme (G2P) transcriber: a canonical pronunciation dictionary, and a dictionary with alternative pronunciations. The dictionary with alternatives takes into account dialectal phonetic and phonological variations, aiming to cope with the variations that appear in the non-read part of the database. The single-entry (canonical) dictionary contains 23 243 words, and the dictionary with alternatives contains 95 778 different pronunciations for the 23 243 lexical entries. This means that each word in the *train* block has 4.12 different pronunciations in average (related lexicon details in section 3.1.6). This fact adds confusion to the acoustic models, since alternatives are used to align the best transcription at a certain stage of the training process, where the best alternative is chosen. That pronunciation alternative is the one that is used until the training process is completed. So, the greater the number of alternatives, the bigger the confusion generated in the acoustic models.

5.2.1 First experiment

To evaluate both groups of HMMs, a phonetic recognition experiment was designed. Here 3 files from each session of the *test* block (a total of 225 files) were tested. The files belong to the subset of phonetically rich sentences, and they have been manually transcribed.

The first step has been a phonetic recognition test of both HMMs created with subset R and subset W. The search graph for the phonetic test has been implemented without any restriction, i.e. in such a way that all phones are equiprobable, and any of them can be followed by any of them. Different Gaussian numbers have been tested in order to see which obtains the best result. Two dictionaries have been used: the one with no alternatives and the one with alternatives.

The phonetic error rates obtained with each acoustic model are shown in Figure 5.2. The lowest error rates are achieved using 32 Gaussians in all the cases. In addition, the use of the dictionary with alternatives does not improve the results, rather the opposite, probably due to the big amount of alternative pronunciations. For numbers of Gaussians smaller than 32, better results are obtained using models trained with subset R, and for numbers of Gaussians greater than 32, using subset W. At the boundary of 32 Gaussians, the best result (12.74 %) is achieved for subset W using the single-entry dictionary, and, using the dictionary with alternatives, the best result (13.35 %) is achieved for subset R.

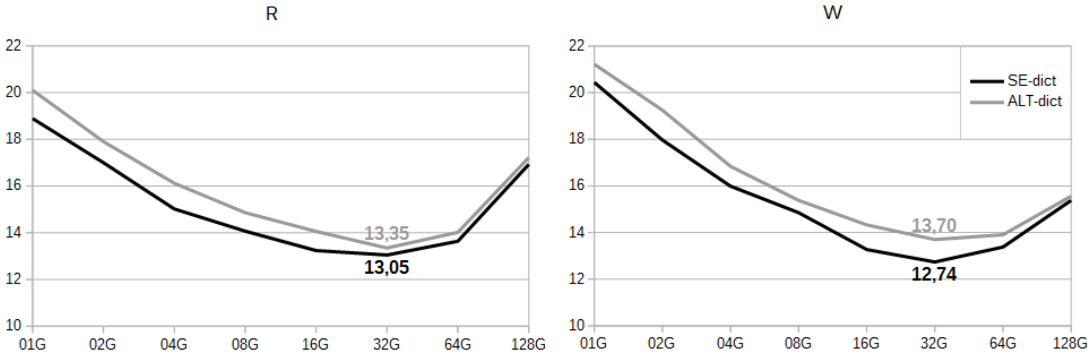


Figure 5.2: 1st experiment: Phonetic error rates (%) for different number of Gaussians obtained in the recognition tests with HMMs trained with subsets R and W using a single-entry dictionary (SE-dict) and a dictionary with alternatives (ALT-dict).

The phone-by-phone breakdown of the best result obtained in the phonetic test shows some noticeable points which deserve to be highlighted. Two significant metrics have been chosen to analyse the result for each phone: the percentage of incorrectly labelled instances (E, i.e. errors) and the percentage of insertions (I) out of the total instances of the phone (see Table 5.1). Considering the two rates, some conclusions have been reached:

- Vowels, laterals and nasals are the phone groups that yield the best results.
- The phone T shows extremely poor results in both rates. This happens because its corresponding phoneme /T/ belongs to Spanish and is not very common in Basque. So, there are very few instances in the database —only a few of them appear in the phonetically rich words part—, and many of them are not correctly pronounced.
- The phone c also shows bad results. The sound corresponding to that phone is not very common in Basque as phoneme —it occurs in central dialects, mostly as an allophone of /t/ in very specific locations—, and consequently many instances in the database are not correctly pronounced. The confusion matrix shows that it is primarily replaced by tS, which corresponds to another allophone of /t/ as well.
- The phones gj, jj and L mingle among them. The confusion matrix shows that many jj and gj are labelled as L; in addition, gj and jj are allophones of the same phoneme which occur at different locations. The *Graph Manager* does not take into account which of them has to be used in each case.
- The phone S has a poor E rate. The corresponding phoneme does not exist in Spanish, and therefore many instances in the database are not correctly pronounced. The confusion matrix shows that it is mainly labelled as s, which corresponds to the sound that Spanish speakers usually use when uttering this phoneme.
- The unvoiced plosive phones p, t and k have good E rates, but poor I rates. This means that they are overused, mainly in the place of glitches and non-speech events. The most evident case is the p.
- The phone f has a very high insertion rate. It mostly appears in noisy non-speech segments.

Table 5.1: Percentages of incorrectly labelled instances (E) and insertions (I) of each phone in the phonetic recognition test giving the best result in the 1st experiment.

	a	e	i	o	u	c	p	t	k	l	r	rr	ts‘	ts	tS
E	3.9	8.7	5.6	15.2	0.9	72.3	8.5	2.7	4.2	11.9	14.4	9.7	18.4	64.1	35.1
I	13.07	16.15	16.91	13.10	10.39	51.06	58.12	26.24	23.01	18.44	13.18	35.00	41.23	14.06	16.22

	m	n	J	L	jj	gj	b	d	g	f	x	T	s‘	s	S
E	12.5	6.6	10.4	38.9	61.9	67.7	15.0	28.9	21.6	13.0	29.2	75.0	18.8	28.0	66.7
I	26.32	15.03	8.33	72.22	7.94	58.06	15.27	23.54	26.80	69.57	10.42	137.50	26.92	4.38	33.33

As explained before, many errors are caused by phonetic transcription ambiguities in the system dictionary. Some phones are differently pronounced, and that is the reason why a dictionary with alternatives was considered. The large amount of pronunciations so obtained adds confusion to the system, and the phonetic results show that worst results are obtained in this way.

5.2.2 Second experiment

A solution for the system to choose the correct transcription from the various alternatives is to manually correct a small amount of transcriptions. This allows the system to correctly align the HMMs at the initial stage of the training process, and minimise errors at the next stage. Initially 12 and then 25 sessions (the whole sessions) were manually corrected. The correction included the phonetic transcriptions as well as the acoustic event labels. The corrected sessions were those corresponding to the first consecutive 12 sessions in the database (called *M12* hereafter) and the first consecutive 25 sessions (*M25*, hereafter).

In this scenario, a second test was carried out, to see what the consequences of using *M12* and *M25* are. This second test consists in developing new HMMs using subsets *R* and *W* with the manually corrected sessions, and performing a new recognition test. Figure 5.3 shows, as a example, the different subsets of the database involved in the test *R*+*M12*. Notice that one session corresponds to one speaker.

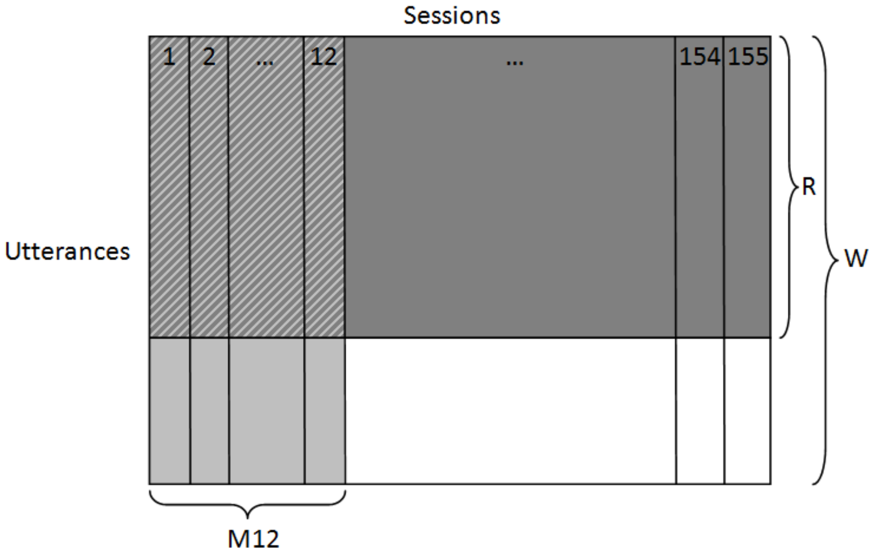


Figure 5.3: Example of the subsets in the *train* block of the *Basque Speecon-like* database, used to train HMMs using *R*+*M12*.

The tested files are the same as those used in the previous test. The PERs obtained for different number of Gaussians are shown in Figure 5.4, along with the data from Figure 5.2 (in black) to get a better picture of the results.

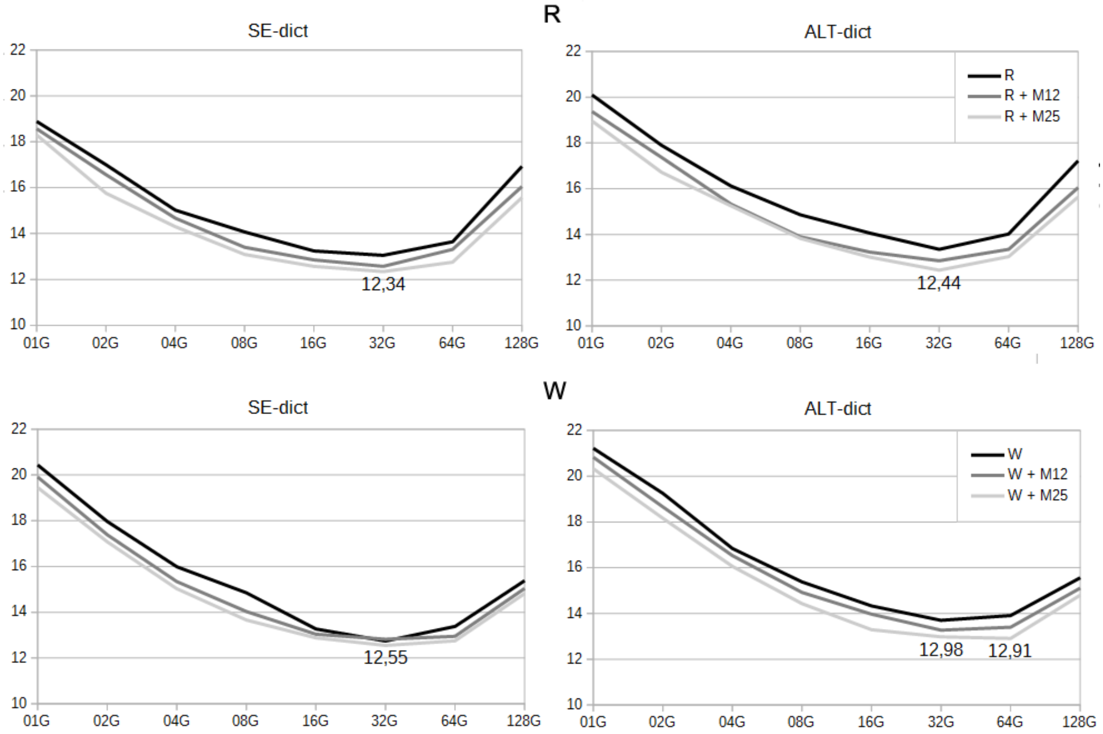


Figure 5.4: 2nd experiment: Phonetic error rates (%) for different number of Gaussians obtained in the recognition tests with HMMs trained with subsets R (top) and W (bottom) without and with manually corrected sessions (—, *M12* and *M25*) using a single-entry dictionary (SE-dict) and a dictionary with alternatives (ALT-dict).

The results show that, as expected, error rates are lower as more manually corrected sessions are considered. In this test, in almost all the cases, the best results are obtained for subset R (and 32 Gaussians). Nevertheless, the purpose of this test was to get a better alignment in the creation of the HMMs when using the dictionary with alternatives, and the best result is still obtained using the single-entry dictionary. However, the best results of the experiment with the single-entry dictionary (% 12.34) and the experiment with the dictionary with alternatives (% 12.44) have come closer. Apparently, the use of manually corrected sessions has a bigger impact on the results of the experiments using the dictionary with alternatives; likewise the impact is bigger on HMMs trained with subset R.

5.2.3 Third experiment

Keeping in mind that the main goal is getting a good alignment of the pronunciation alternatives when creating HMMs, a third test has been devised. Here, different ways of creating HMMs have been considered. The HMM training process comprises several stages, starting from monophones and ending with triphones. Typically, at the first stages, the single-entry dictionary is used, then, at an intermediate stage, a forced alignment recognition test is performed using the dictionary with alternatives, in order to choose the best phonetic transcription corresponding to each word. From that stage on, the transcriptions obtained from the alignment process are used, ensuring thus the use of the best transcription alternative for each word. The way that these initial HMMs are estimated is crucial for achieving good transcription alignments; so, the more accurate the initial transcriptions, the better the final HMMs will be.

Taking the above into account, the third test consists in considering different subsets for the first stages and the subsequent stages of the training process (using the dictionary with alternatives in all of them). Figure 5.5 shows the results for *M25* (*M12* has been obviated). Note the nomenclature used: "*M25 — R+M25*" means that at the first stages of the training process only the first 25 sessions have been used (those corresponding to *M25* subset only), and then, after the transcription alignment, the union of subsets *R* and *M25*.

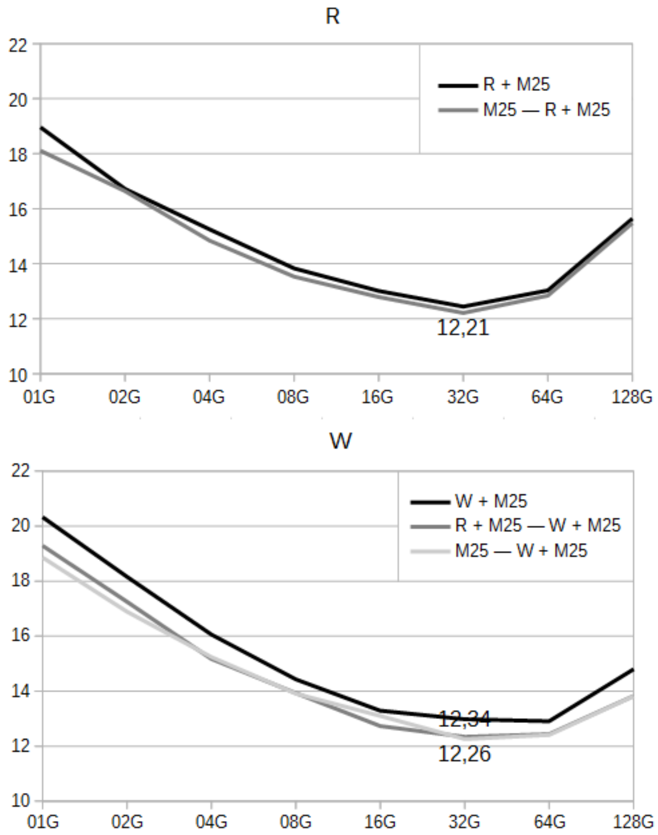


Figure 5.5: 3rd experiment: Phonetic error rates (%) for different number of Gaussians obtained in the recognition tests using phased trained HMMs —in comparison with not phased (in black)— using the dictionary with alternatives (for *M25*).

With this phased training technique, the best outcomes so far have been obtained: 12.21 % and 12.26 %, belonging to subset *R* ("*M25* — *R*+*M25*") and subset *W* ("*M25* — *W*+*M25*") experiments, respectively. Besides, a result of 12.34 % (equal to the best result of the 2nd test) has been also obtained for "*R*+*M25* — *W*+*M25*". Although the differences between the best results of the 2nd test (with the single-entry dictionary) and this 3rd test are very small, it becomes evident that a better alignment has been finally achieved using the dictionary with alternatives and training the HMMs initially with a subset containing very few transcription errors and then with a bigger subset.

To better understand the results obtained in this third test, two illustrative charts are shown in Figure 5.6. Both charts show absolute differences obtained using a dictionary with alternatives with respect to the single-entry dictionary results, using a different amount of manually corrected transcriptions (*0*, *M12* or *M25*). The chart in the left refers to the experiments carried out using subset *R*, and the one in the right refers to the ones using subset *W*. In each chart two diagrams are shown: one corresponding to results obtained using not-phased trained HMMs (left) and the other to phased trained HMMs (right).

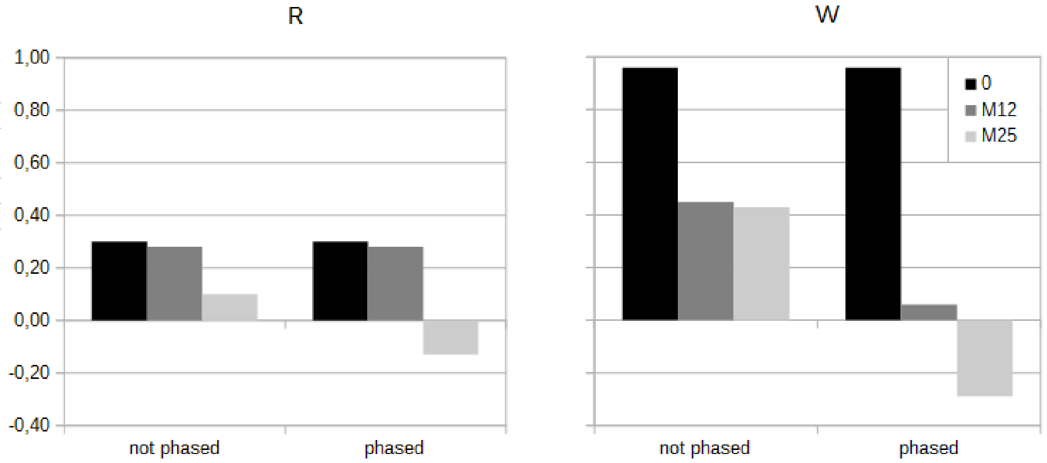


Figure 5.6: Absolute PER differences of experiments using a dictionary with alternatives and phased and not-phased trained HMMs (32 Gaussians) with respect to the single-entry dictionary, for different amounts of manually corrected transcriptions (left: results using subset *R*; right: using subset *W*).

The diagrams show that, either for subset *R* ("*M25* — *R*+*M25*") or for subset *W* ("*M25* — *W*+*M25*"), the use of alternatives with not-phased trained models do not provide better results than those obtained using the single-entry dictionary (although, as expected, the greater the amount of manually corrected transcriptions, the better the results). However, better outcomes are achieved using alternatives with phased trained models; an improvement is even obtained for *M25*. Note that the results obtained with no manually corrected transcriptions (black bars in the charts) are the same at phased and not-phased modes, since the same subset would be used in each training phase.

In order to see the impact of the use of phased-trained models on the results at phone level, a phone-by-phone breakdown has been also generated for the HMMs achieving the best result ("*M25 — R+M25*", dictionary with alternatives, 32 Gaussians). Results can be seen in Table 5.2.

Table 5.2: Percentage of incorrectly labelled instances (E) and insertions (I) of each phone in the phonetic recognition test giving the best result in the 3rd test.

	a	e	i	o	u	c	p	t	k	l	r	rr	ts‘	ts	tS
E	4.1	7.5	6.5	5.8	11,3	61.7	11.1	3.6	4.1	13.7	17.8	7.5	16.1	57.6	35.1
I	10.57	13.96	14.03	10.98	8.75	10.64	41.03	15.74	18.48	13.71	6.02	22.96	28.57	15.15	18.92
	m	n	J	L	jj	gj	b	d	g	f	x	T	s‘	s	S
E	15.0	6.8	10.4	55.6	54.8	33.5	16.5	17.6	20.6	4.3	6.2	62.5	21.2	18.9	58.3
I	20.26	10.41	29.17	55.56	130.65	6.45	11.85	16.84	26.60	47.83	11.46	137.50	15.87	4.84	50.00

To better interpret the phone-level results, a comparative chart has been created: Figure 5.7. The figure shows the results of Table 5.1 (grey bars) together with those of Table 5.2 (black bars) phone by phone: E ratio is shown in the chart of the top and I ratio in the chart of the bottom. Generally speaking, lower E ratios are obtained with the 3rd test HMMs, although in some cases (phones u and L) E increases significantly. Regarding the I, better results have also been obtained in general, with the remarkable exception of the phone jj, whose insertion rate has grown dramatically.

The anomaly of the phone jj is a consequence of an inappropriate alignment. In some dialects of Basque an extra phone is added when a noun or adjective ends in $/i/$ and the article -*a* is appended (the extra phone is added between them). Different sounds are added depending on the dialect ($/jj/$ is the corresponding to the central dialects), and all of them are included in the pronunciation dictionary with alternatives. Since in the manually corrected subset this addition does not even once occur, the decoder has no enough data to decide whether it is one or the other. So, in the alignment process many false additions have been inserted. The inclusion of some manually corrected dialectal data would be enough to solve this problem. Nonetheless, it will be done in future works, since in principle it would only concern a particular triphone.

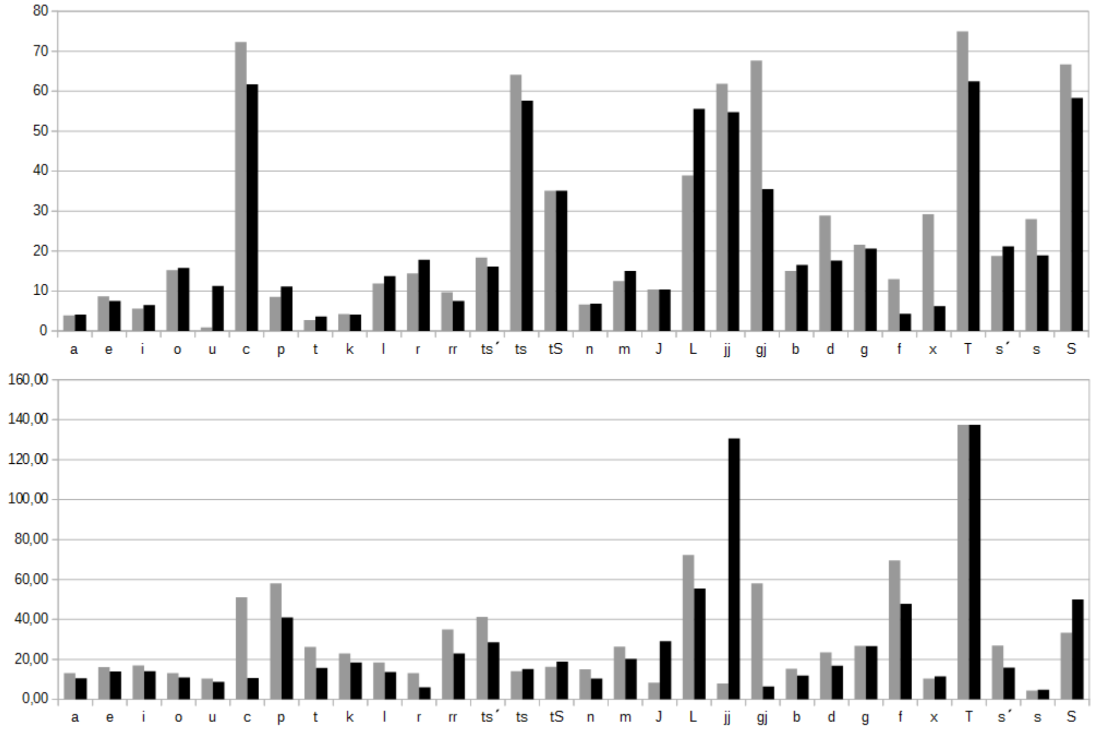


Figure 5.7: Comparative chart of phonetic recognition results of the HMMs giving the best results in 1st test (grey columns) and 3rd test (black columns). At the top, incorrectly labelled instances (E); at the bottom: insertions (I).

5.3 Feature normalisation: CMVN

Parameter normalisation is essential to create robust acoustic models and cope with audio signals captured by different microphones. Although the initial experiments, as explained in Chapter 6, have been carried out with a single PC and using the same headphones for all the students (without parameter normalisation), the web implementation of the system requires some kind of normalisation since students will use each his or her own equipment.

In order to compensate for the differences of the incoming signals (channels, background noise, etc.), the most common practice is to perform Cepstral Mean and Variance Normalisation (CMVN) of the extracted features. This is the approach chosen for this thesis, and so new acoustic models have been trained and tested using normalised MFCCs. For further details, the basis of CMVN is described in Chapter 9.

As explained in [135], the MFCC mean vector conveys the spectral characteristics of the current microphone and room acoustics. In the limit, when $N \longrightarrow \infty$ for each utterance, we should expect means from utterances from the same recording environment to be the same. Therefore, as each speaker of the *Basque Speecon-like* database has been recorded under the same acoustic conditions, the MFCCs corresponding to the audio signals of the database have been normalised by speaker. Mean and variance vectors have been calculated over all the data corresponding to the same speaker (316 utterances). Then, the new normalised MFCC files were obtained and the new HMMs were trained.

5.3.1 Experiments

The three tests performed in section 5.2 have been repeated using normalised HMMs in order to see the impact of this technique. Normalisation is expected to draw good results when the tested files present differences in equipment, environment or background noise compared with those used to develop the HMMs. Since the tested files belong to the same database as the training data, this scenario does not appear to be the best to take advantage of normalisation. However, it can shed light on the behaviour of the normalised HMMs. The results obtained in the new three tests can be seen in Figure 5.8 (1st experiment), Figure 5.9 (2nd exp.) and Figure 5.10 (3rd exp.).

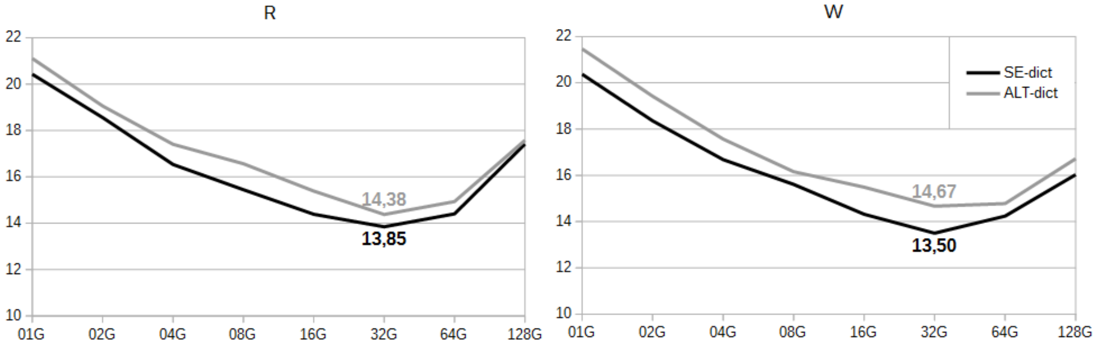


Figure 5.8: 1st experiment repeated using CMVN.

Results are slightly worse than those obtained without normalising parameters. The best result of all, of 12.45 %, is obtained using subset *R* with *M25* and the single-entry dictionary. Besides, the evolution across the number of Gaussians seems to have moved, since more experiments show now a minimum peak of error at 64 Gaussians, mainly those that include the *W* subset.

Additionally, experiments using only subsets *M12* and *M25* have been performed. They have not been included so far because, although they are completely correctly labelled, they contain too few data to properly train HMMs. Thus, the same phonetic experiments with and without CMVN have been performed for subsets *M12* and *M25*, and contrary to what happened so far, results are better with CMVN. Figure 5.11 shows graphically the results obtained for these two cases along with *R*, *R*+*M12* and *R*+*M25*, in order to see different behaviours of HMMs trained using different subsets.

While for subset *R* similar results are obtained, and worse results for *R*+*M12* and *R*+*M25*, an absolute improvement of about 1 % has been achieved for subsets *M12* and *M25*. This shows that parameter normalisation may be beneficial when the amount of

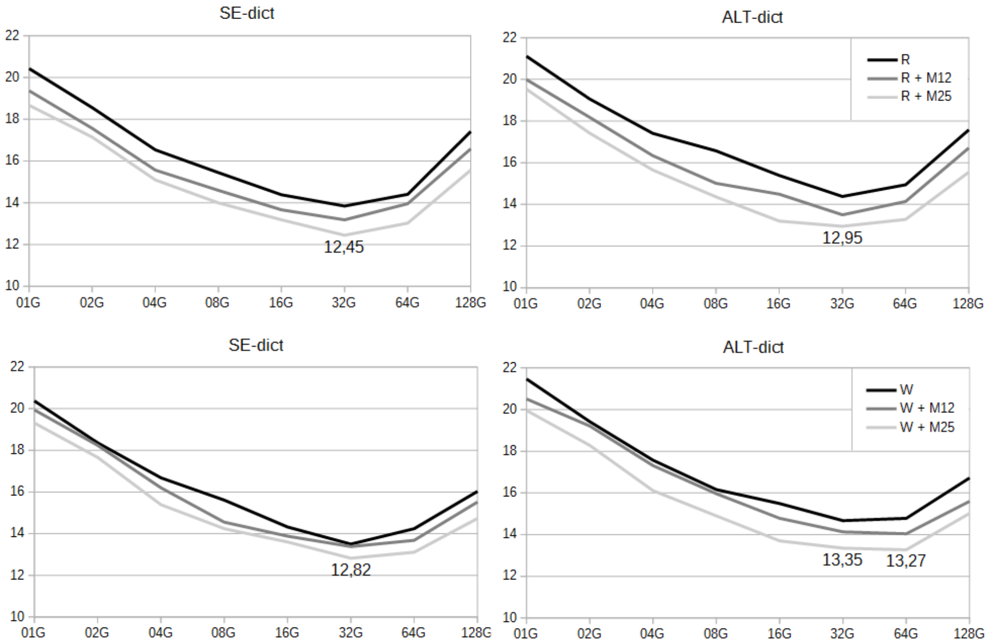


Figure 5.9: 2nd experiment repeated using CMVN.

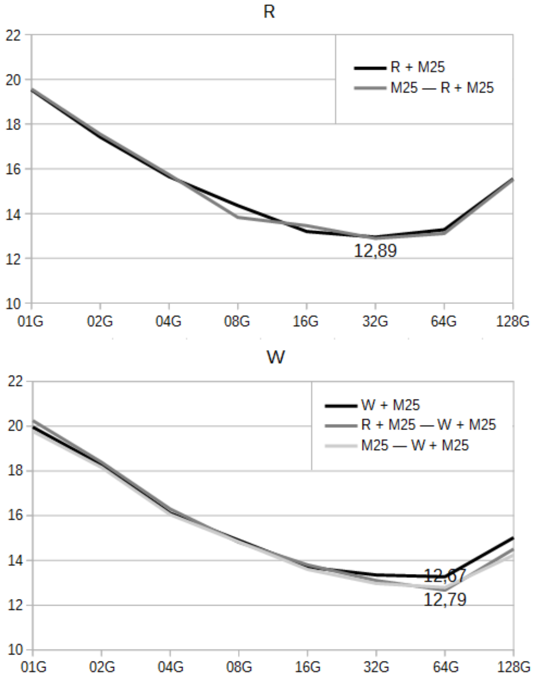


Figure 5.10: 3rd experiment repeated using CMVN.

errors in the transcriptions is low.

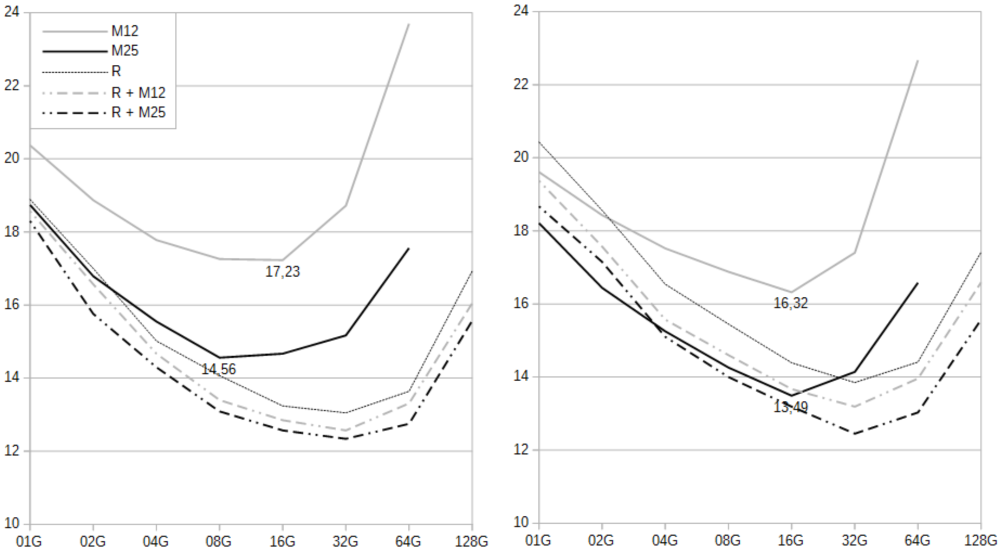


Figure 5.11: Phonetic error rates (%) of the HMMs trained using subsets *M12* and *M25*, along with other subsets, without (left chart) and with (right chart) CMVN.

Concerning the results of CMVN of using a dictionary with alternatives with respect to using the single-entry dictionary, a double chart similar to Figure 5.6 has been generated (see Figure 5.12). Unlike in Figure 5.6, no better results than the ones obtained using the single-entry dictionary are achieved using dictionary alternatives, although results

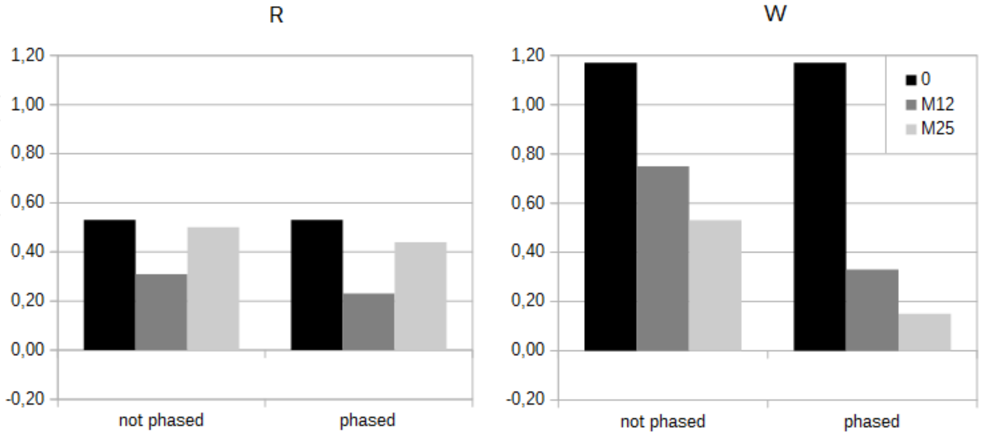


Figure 5.12: Absolute PER differences of CMVN experiments using a dictionary with alternatives and phased and not-phased trained HMMs (32 Gaussians) with respect to the single-entry dictionary, for different amounts of manually corrected transcriptions (left: results using subset *R*; right: using subset *W*).

get better as more manually corrected transcriptions are being used. It is noticeable that for subset *R* the increase of the error is greater for *M25* than for *M12*. Nevertheless, the results are better for *M25*.

5.4 Testing the channel mismatch

The effects of CMVN are not noticeable when testing signals recorded under the same acoustic conditions as those used for training the HMMs. Since our tool is located in a server and each user will use a different microphone in a different environment, there will be a channel mismatch between the training data and the signals that the decoder will receive. In this scenario the advantages of cepstral normalisation are supposed to become apparent.

To account for this issue, a new test has been carried out: the HMMs trained as described in the previous section (both non-normalised and normalised) have been used to test audio signals whose recording channel is different. To that end, files belonging to the block recorded through a *desktop* microphone (at 1 *m* distance) in the *Basque Speecon-like* database have been used. The content of these new test files is the same as the content of the files used previously (225 files, 3 files per speaker —see section 5.2.1—); the only difference is that they have been recorded through a desktop microphone instead of a head-mounted microphone.

Table 5.3 shows some results obtained in these tests (PER (%)). Although all the differently trained HMMs of the previous tests have been used, only the most representative data have been chosen to be shown in the table: the test performed with the HMMs trained with 32 Gaussians and including 25 manually corrected sessions (*M25*). The files have been tested with and without CMVN.

Table 5.3: Phonetic error rates (%) of the phonetic recognition tests performed on the *desktop* sub-corpus of the *Basque Speecon-like* database, using HMMs trained with the *close-talk* sub-corpus, with and without CMVN (32 Gaussians).

<table>
<tr><th colspan="3"></th><th>Without CMVN</th><th>With CMVN</th></tr>
<tr><td rowspan="2">SE-dict</td><td><i>R</i></td><td><i>M25</i></td><td>26.56</td><td>21.87</td></tr>
<tr><td><i>W</i></td><td><i>M25</i></td><td><b>25.92</b></td><td><b>21.83</b></td></tr>
<tr><td rowspan="5">ALT-dict</td><td><i>R</i></td><td><i>M25</i></td><td>27.21</td><td>22.41</td></tr>
<tr><td><i>W</i></td><td><i>M25</i></td><td>26.37</td><td>22.55</td></tr>
<tr><td colspan="2"><i>M25</i> — <i>R+M25</i></td><td>27.32</td><td>22.15</td></tr>
<tr><td colspan="2"><i>M25</i> — <i>W+M25</i></td><td>26.93</td><td>22.09</td></tr>
<tr><td colspan="2"><i>R+M25</i> — <i>W+M25</i></td><td>27.18</td><td>21.85</td></tr>
</table>

Results underline the impact of channel mismatch. Without normalisation, the best PER obtained is 25.92 % testing *desktop* sub-corpus files (channel mismatch), while 12.21 % was obtained testing *close-talk* files (similar channels). So, phonetic error rates under channel mismatch conditions are a little bit higher than twice the error rates under similar channel conditions. With CMVN, the best PER obtained is 21.83 % testing *desktop* files and 12.45 % testing *close-talk* files. In this case, phonetic error rates under channel mismatch conditions do not reach the double, but are still rather high. However, a global improvement of about 17 % has been achieved with CMVN taking into account all the tests under mismatch conditions, while nearly no impact was obtained when files recorded through similar channels were tested.

5.5 Conclusions

In this section we have trained and tested HMMs created in different ways. Using the pronunciation dictionary with alternatives has proved to cause a detrimental effect, since a large amount of pronunciation alternatives has to be taken into account. That is why phased HMM training procedures have been performed. The aim is to achieve a good pronunciation alignment when using a dictionary with alternatives, and, for this purpose, a small part of the database has been manually corrected, expecting that it would act as a guide for a proper alignment. These corrections along with a phased training have obtained the best result: a PER of 12.21 % (process "*M25 — R+M25*" using the dictionary with alternatives and 32 Gaussians). Although good results have been obtained, a better improvement was expected initially. One reason of not obtaining the expected improvement is the high insertion rate of the phone *jj*, a not frequent phone in Basque, which is not trained properly since no occurrence of this phone appears in the manually corrected subsets.

When testing the HMMs trained using normalised parameters, almost the same results (slightly worse) have been obtained, probably because the effects of normalisation cannot be seen testing files that have been recorded using the same equipment and often the same environment. However, the benefits of using CMVN have become apparent in the phonetic recognition test performed using only manually corrected data.

In addition, all the tests have been repeated to check how HMMs behave when audio files recorded in different environments and using different microphones are processed. For that purpose, files recorded by means of a microphone located at a distance of 1 m have been tested. Without CMVN, the resulting error rate increase is even higher than 100 %. With CMVN, a smaller error rate increase is obtained, of about 70 %. Hence, CMVN has a noticeable impact on the results, but not as big as expected.

Examining the results, two different sets of HMMs have been chosen for the design of the final system:

- **Without CMVN**: the HMMs created following the process "*M25 — R+M25*" have been considered, using the dictionary with alternatives, and 32 Gaussians.
- **With CMVN**: the HMMs created following the process "*R+M25*", using the single-entry dictionary, and 32 Gaussians.

Noteworthy are the PERs obtained in this chapter, which are significantly smaller than the ones currently obtained for the TIMIT database (16.5 %, as commented in the introduction of this chapter), even without using state-of-the-art techniques. Logically, the differences between both databases have a deep impact on the outcomes.

CHAPTER 6

Early experiments and the initial system

6.1 Introduction

At the beginning of this work, the possibility of implementing the system as an on-line server had not even been discussed. Audio capture features were not implemented in browsers, and compatibility with plugins or external applications such as *Flash* or *Java applets* was quite limited. Consequently, *AhoSR* was initially designed to be run in local mode.

The first prototype of *AhoSR* was ready in 2009, but the very first verification experiments for language learning started in 2011. At that time, our *AhoSR* system did not include neither Voice Activity Detection (VAD) nor Cepstral Mean and Variance Normalization (CMVN) (these features have been implemented taking advantage of this thesis). The *Basque Speecon-like* database, the one used to train the HMMs, lacked of phonetic transcriptions and a long work was foreseen to cope with that issue.

In that context, the models we were using for ASR were HMMs created using the *R* subset of the database (see section 5.2) leaving aside the spontaneous speech part to guarantee a minimum amount of transcription errors and disfluencies (see section 3.1.3). Although much data was discarded, preliminary tests were designed using the *R* subset while a solution for managing the dialectal variations was being sought.

As the starting point of all this work, we tried to see whether those HMMs created for ASR were useful to train confidence scores (such as GOPs) to discern between phonemes. The first experiments were initially carried out in the laboratory, but the need of getting results in a real-world environment was also foreseen. All in all, a software called *AhoSR_L2* was developed in C++, which was executed in local mode.

In this chapter, we present the method which was initially chosen to assess whether an input phone is correctly pronounced or not, based on GOP scores. Actually, the *Basque Speecon-like* database is not a CAPT database, and so it is not designed to include recordings where phonemes are incorrectly pronounced by second language speakers. That led us to devise a method to obtain the GOP distribution of the incorrectly pronounced phonemes. This method is explained in the next section (section 6.2).

Section 6.3 describes a preliminary evaluation of the pronunciation evaluation task, carried out under laboratory conditions. Then a description of the software *AhoSR_L2* is given in section 6.4. The design of the linguistic exercises and the evaluation is presented next in section 6.5, and finally some conclusions are outlined.

6.2 Phone scoring: GOPs and decision thresholds

6.2.1 Basic GOP algorithm

The GOP score of a phone q_i is computed as the duration-normalised log of its posterior probability $p(q_i|O_i)$ over the acoustic segment O_i (i denotes the phone index inside the word), as defined in equation (6.1) [47].

$$GOP\left(q_i\right) = \frac{1}{T_i} logp\left(q_i|O_i\right) = \frac{1}{T_i} log\left[\frac{p\left(O_i|q_i\right)p\left(q_i\right)}{\sum_{j=1}^{N} p\left(O_i|q_j\right)p\left(q_j\right)}\right] \tag{6.1}$$

where T_i is the number of frames that the given segment lasts and N is the number of phones in the phone set. Assuming that all phones are equally likely $(p(q_i) = p(q_j))$ and that the sum in the denominator can be approximated by its maximum, the basic GOP measure becomes:

$$GOP\left(q_i\right) \approx \frac{1}{T_i} log\left[\frac{p\left(O_i|q_i\right)}{p\left(O_i|q_{j_{max}}\right)}\right] \tag{6.2}$$

where j_{max} is the phone model index that gives the highest likelihood for that segment. The acoustic segment boundaries and the corresponding likelihoods are determined from the Viterbi alignment. Firstly, the numerator of equation (6.2) is computed in forced alignment mode in which the sequence of phone models is fixed; secondly, the denominator is determined using an unconstrained phone loop. Thus, the denominator score is determined by simply summing the log likelihood per frame over the duration of O_i. In practice, this often means that more than one phone in the unconstrained phone sequence has contributed to the computation of $p(O_i|q_{j_{max}})$.

6.2.2 Setting the thresholds

Theoretically, two GOP distributions are needed to decide whether a phoneme is correctly or incorrectly pronounced using GOP scores: on the one hand, the GOP distribution of the correctly pronounced instances, which can be obtained from any ASR database (the *Basque Speecon-like* database, in our case) computing the GOPs in forced alignment mode. On the other hand, the GOP distribution of the incorrectly pronounced instances. This kind of data is more difficult to obtain.

Different ways have been described in the literature to calculate the GOP thresholds of mispronounced phonemes:

1. **Empirical calculation**: The threshold for a phone q_i can be set in terms of the mean μ_i and variance σ_i of all the GOP scores for the phone q_i in the training data, as indicated in equation (6.3).

$$T_{q_i} = \mu_i + \alpha\sigma_i + \beta \tag{6.3}$$

   where α and β are empirically determined scaling constants. In [46], these values are estimated to be: $0.8 < \alpha < 1.3$ and $-1.0 < \beta < 2.0$, yielding thresholds on a similar scale as the global threshold, but adapted to individual phones. The assumption is that averaging the native GOP scores will reduce the effect of errors in the phone recogniser.

2. **Learning from human judges**: A reasonable target for an automatic pronunciation system is to perform as well as a human judge. One way to approximate human performance is to learn from human labelling behaviour, as explained in [47]: Let $c_s(q_i)$ be the total number of times that a phone q_i uttered by speaker s is marked as mispronounced by one of the human judges in the training database. Then, a threshold can be defined by averaging the normalised rejection counts over all speakers (see equation (6.4)).

$$T_{q_i} = log\frac{1}{S}\sum_{s=1}^{S}\left(\frac{c_s(q_i)}{\sum_{j=1}^{N} c_s(j)}\right) \tag{6.4}$$

   where N is the total number of distinct phones and S is the total number of speakers in the training set. Due to the normalisation, the average counts are in the range of 0 to 1, so that the resulting logarithmic values yield thresholds on a similar scale to those defined in equation (6.3).

3. **Explicit error modelling**: Pronunciation errors can be grouped into two main error classes. The first class contains individual mispronunciations which occur if a student is not familiar with the pronunciation of a specific word. The second class consists of substitutions of sounds in the target language which do not exist in the native language. This latter error type is known as *systematic mispronunciation.*

   Since the GOP method does not employ models for the phones of all the students' native languages, incorrect acoustic modelling of the non-native speech will occur in the case of systematic mispronunciations. Therefore, the detection of these errors might be improved if knowledge of the native tongue of the learner can be included in the GOP scoring. For this purpose a recognition network can be implemented incorporating both correct pronunciation and common pronunciation errors in the form of error sub-lattices for each phone, using the phone model sets of both the target and the source language. This method has the disadvantage of having to obtain acoustic models for every source language.

4. **Simulated or artificial errors**: Another way to obtain phone thresholds is using artificial data. As explained in [46], artificial data is that created by manipulating the pronunciation dictionary so that the pronunciations are changed to contain different phones. For instance, all occurrences of the sound /*aa*/ could be changed to /*iy*/ and so forth. Thus, speech data with known locations of pronunciation errors can be created. Experiments in that work show that monophone models outperform triphone models, obtaining a *Scoring Accuracy* of 90% at a *False Acceptance* rate of 8 % for the optimum threshold. For this setup the GOP scoring method seems a viable assessment tool.

For Basque there is no CAPT database available, and so we cannot get the realisations of incorrectly pronounced phonemes. In consequence, we have opted for the solution of creating artificial errors (simulated errors). However, the phonemes in the dictionary have been replaced only with those phonemes belonging to the same acoustic group; actually, when trying to pronounce a phoneme, it is more probable that it is replaced by a "similar" sound (e.g. a vowel by another vowel) instead of a "very different" sound (e.g. a vowel by an unvoiced fricative).

Phone groups

In order to obtain acoustically similar phone groups, a previous clustering work developed in the laboratory was used. As explained in [136], one mixture GMM models were trained per phone using the *Basque Speechdat* database [98]. The phone inventory used was the corresponding to the phone set of SAMPA Basque code[1]. Once the training was carried out, regression trees were used over the acoustic parameters of the GMM models, in order to obtain different clusters. Figure 6.1 shows the dendrogram with the obtained phone clustering.

For that work, a convenient cutting point was chosen (the dashed horizontal line), resulting in eight phone groups (considering vowels as elements of the same group, although strictly speaking each one should define a group). It is remarkable that the resulting groups match almost completely the different articulation modes of the phones of SAMPA Basque code (see section 3.2.2). Note that the phones B, D and G corresponding to the approximant allophones of voiced plosive phonemes were also included in the classification, but in this work they have been discarded, since a single model was created for both phonetic variations; we will see later some consequences of

[1] http://aholab.ehu.eus/sampa_basque.htm

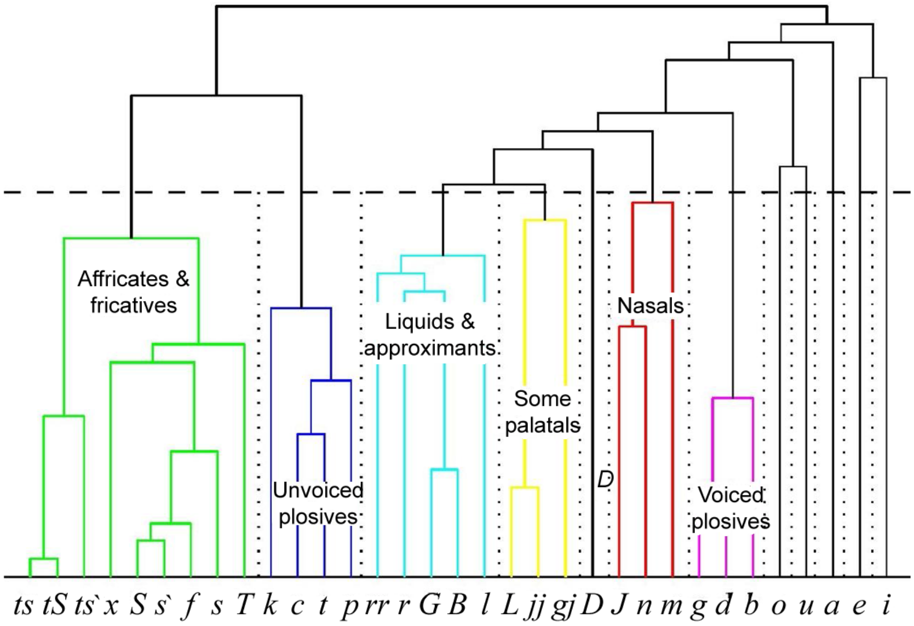


Figure 6.1: Dendrogram of the phone clustering output. Each phone group is represented by a different colour.

this decision. As a result, seven groups remained. In addition, the group of affricates and fricatives was split into two groups, since Basque language presents special characteristics regarding these two groups which will be important to work out with. So it was considered that it is important to train more discerning models to properly distinguish between the phones of one group and the other.

Therefore, eight final groups were defined:

- Vowels: a, e, i, o, u
- Unvoiced plosives: c, p, t, k
- Liquids: r, rr, l
- Affricates: ts‘, ts, tS
- Nasals: m, n, J
- Palatals: L, jj, gj
- Voiced plosives: b, d, g
- Fricatives: f, x, T, s‘, s, S

Having defined the phone groups, the GOP scores for the incorrectly pronounced phonemes were computed, replacing all the phone instances in the dictionary with another one from the same group. Since HMMs were context-dependent HMMs (triphones), the context-dependency of the replaced phones was tuned as well, thus keeping the coherency in the transcriptions.

GOP scores were obtained over all the files of the *train* part of the *Basque Speecon-like* database, in forced alignment mode, and the process was repeated with different error simulations in order to get more data. Then, the histograms of both distributions were calculated for each phone, joining all the scores corresponding to the same phoneme. Thus the verification thresholds were obtained by calculating for each phoneme the Equal Error Rates (EER) derived from the GOP score distribution functions. An example of the distribution of the scores corresponding to the phoneme /*a*/ can be seen in Figure 6.2, with both separate density functions.

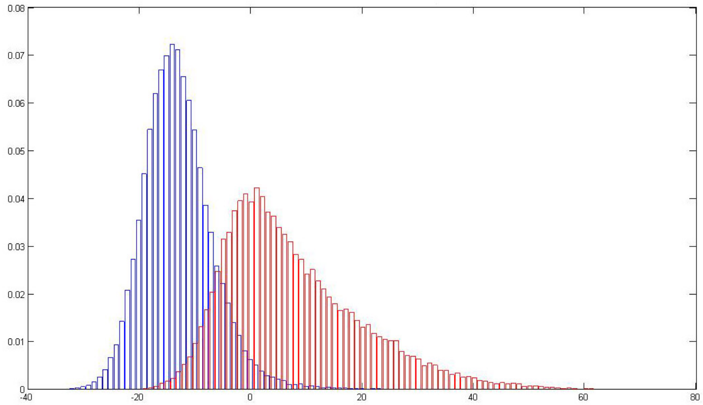


Figure 6.2: Normalised histograms of the GOP scores of phone /*a*/: the blue bars represent the GOP distribution of correctly pronounced phonemes; the red bars represent the GOP distribution of incorrectly pronounced phonemes (simulated errors).

At first glance, it was observed that there were some potentially problematic phones with highly overlapping GOP distributions. This was due to the differences in the pronunciation of some phonemes between native and non-native speakers of Basque in the database, especially in those phonemes that do not exist in the mother tongue of the speaker, as the /*ts*‘/ or the /*s*‘/. Since the HMMs had been trained ignoring this fact, new HMMs were rebuilt, using only the signals corresponding to the native speakers that were born and live in the east area of the country, since in the west area the /*s*‘/ phoneme is not pronounced today ——it merged with the /*s*/——. Thus, 76 eastern Basque speakers out of 155 (those belonging to the *train* part) were used to train the new HMMs. With these new HMMs all the procedure was repeated, and new GOP distributions were obtained. A significant separation between the distributions of the most problematic phonemes was observed. An example is shown in Figure 6.3, where more separate GOP distributions can be observed when using HMMs trained with native speakers’ signals.

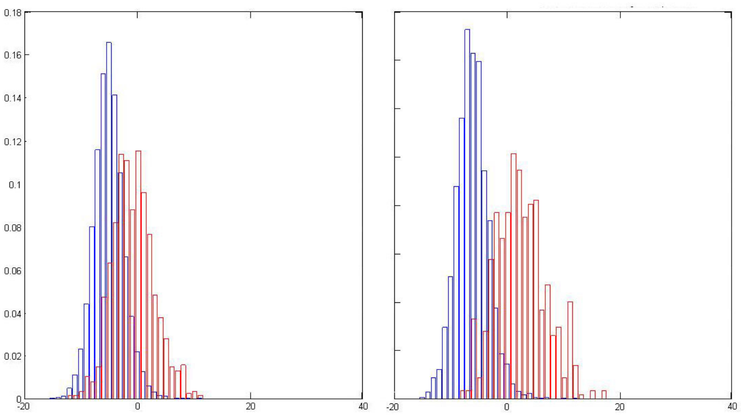


Figure 6.3: Normalised histograms of the GOP scores of the phone ts‘ obtained with HMMs trained using all kind of speakers (left) and using only native speakers' data (right)

Nevertheless, problematic phonemes still exist, as is the case of phone $/ts/$, which in western dialects of Basque is pronounced as $/ts$‘$/$ and is getting closer to $/tS/$ in central dialects. The distribution pairs were not separated using the new HMMs, as it is shown in Figure 6.4.

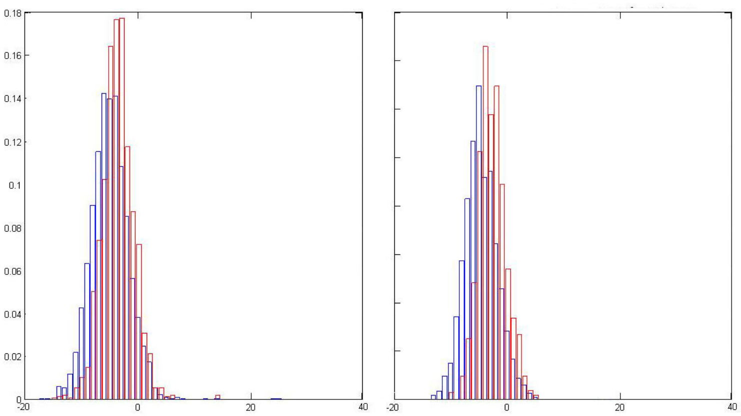


Figure 6.4: Normalised histograms of the GOP scores of the phone ts obtained with HMMs trained using all kind of speakers (left) and using only native speakers' data (right)

The EERs obtained using only native speakers' audio signals to train HMMs are shown in Table 6.1. While some phones have a low EER value, others have higher values that should be improved. In the case of phones corresponding to voiced plosive phonemes, it may be due to the fact that, as explained before, two different realisations of the same phoneme have been merged in a single phone. In the case of the sibilants, two problems arise: on one hand, the $/s`/$ is pronounced as $/s/$ in some areas of the Basque Country, and, on the other hand, the $/ts/$ is pronounced as $/ts`/$ in some areas of the Basque Country and $/tS/$ in some others. So the corresponding HMMs have not been properly trained. Note that the phone gj has no result, due to the scarcity of data available of its corresponding phoneme.

Table 6.1: EER values for each phone, computed using the simulated errors method.

Phone	EER	Phone	EER	Phone	EER	Phone	EER	Phone	EER
$/a/$	14.03	$/p/$	18.30	$/ts`/$	10.61	$/L/$	19.14	$/f/$	12.85
$/e/$	18.64	$/t/$	20.54	$/ts/$	36.74	$/jj/$	32.05	$/x/$	5.10
$/i/$	07.99	$/k/$	22.34	$/tS/$	27.85	$/gj/$	-	$/T/$	24.62
$/o/$	15.53	$/r/$	17.11	$/m/$	15.66	$/b/$	30.09	$/s`/$	34.82
$/u/$	10.92	$/rr/$	14.66	$/n/$	38.45	$/d/$	24.88	$/s/$	16.94
$/c/$	26.18	$/l/$	17.66	$/J/$	13.70	$/g/$	28.93	$/S/$	20.47

Part of the work explained in this section was carried out during my research stay in the Institute of Acoustics and Speech Communications, Technical University Dresden (Germany) in Dresden (Germany) in 2012. For further details, see the article published as a result [68].

6.3 Preliminary evaluation

The measure selected to assess the results is the widely used SA (Scoring Accuracy) coefficient, which is computed as in equation (6.5).

$$SA\,(\%) = \left(\frac{CA + CR}{CA + CR + FA + FR}\right) \cdot 100 \tag{6.5}$$

where CA: Correctly Accepted units; CR: Correctly Rejected units; FA: Falsely Accepted units; FR: Falsely Rejected units.

6.3.1 Phone-level tests

Two different experiments were performed, using different sets of recordings, in order to evaluate the designed strategy. In both tests, the HMM set used was the one trained with eastern native speakers. The GOP score for each phone was calculated using a forced alignment procedure and compared with the EER thresholds.

- **Test 1**: The tests were performed choosing the native speakers who were born and live in the east area of the Basque Country out of the *test* block of the database (13 speakers out of 75, see section 3.1.4). The transcriptions corresponding to these speakers were not checked, but we assumed that there was no error and that all the speakers pronounced all the phonemes properly. So, the phones that the system evaluates as correct were supposed to be really correct (CA), and the phones evaluated as incorrect were supposed to be false rejects (FR).

- **Test 2**: It was performed using the recordings of the speakers with "low level" skills in Basque, 25 in total, out of 75. Even though it is not possible to predict a priori if a phoneme had been correctly or incorrectly pronounced, the tests were carried out considering no linguistic knowledge and labelling, using the same set as in Test 1, in order to see at first sight how the performance of the system could be for "low-level" students.

The results of four representative phones are shown in Table 6.2: two vowels ($/a/$ and $/u/$), giving good results; and two sibilants ($/ts`/$ and $/s`/$), corresponding to phonemes that do not exist in Spanish. The SAs of phones $/a/$ and $/u/$ have a small variation from one test to the other, which means that native and non-native speakers pronounce them similarly. However, the SAs of $/ts`/$ and $/s`/$ fall significantly, especially in the case of the $/ts`/$, which means that the phonemes corresponding to these phones are pronounced in quite a different way.

Table 6.2: Number of realisations and SA for the phones $/a/$, $/u/$, $/ts`/$ and $/s`/$ in tests 1 and 2.

		$/a/$	$/u/$	$/ts`/$	$/s`/$
Test 1	#Realis.	5 524	1 937	750	1 317
	SA (%)	86.22	89.67	83.73	74.26
Test 2	#Realis.	9 923	3 481	1 438	2 469
	SA (%)	84.06	87.33	41.59	49.49

Finally, another experiment was devised:

- **Test 3**: An experiment on the phones that showed the worst SAs in Test 2. The realisations of the corresponding phonemes in the subcorpus of "low-level" skills in L2 were manually labelled as correctly or incorrectly pronounced. In the case of the phoneme $/ts`/$ there were 813 realisations, 375 being labelled as correct and 438 as incorrect. For the phoneme $/s`/$, 1 348 realisations were labelled; 720 being labelled as correct and 628 as incorrect. Considering these labels, the GOPs were calculated again, obtaining the results of Table 6.3.

We can see that the SAs of Test 3 (Table 6.3) are now closer to the ones obtained in Test 1 (Table 6.2) compared with the results of Test 2. This confirms the existence of incorrectly pronounced phonemes in the database. The results of the phone $/ts`/$ show that the distribution separation obtained training HMMs with only native speakers'

Table 6.3: Results of comparing the automatically generated labels with the manually assigned labels, for phones ts‘ and s‘.

Test 3	/ts‘/	/s‘/
CA (%)	32.84	33.97
CR (%)	43.67	29.15
FA (%)	10.21	17.43
FR (%)	13.28	19.43
SA (%)	76.51	63.13

data (shown in Figure 6.3) has been useful to calculate decision thresholds. The less accurate results for the phone /s‘/ can be interpreted taking into account that nowadays the corresponding sound /s‘/ does not exist in a big area of the Basque Country, so that even some native speakers of the eastern side may be influenced by this fact.

From a general point of view, the results show that our *AhoSR_L2* system was able to detect both correctly pronounced phonemes in an utterance and incorrectly pronounced ones, focusing especially on phonemes that do not exist in the L1 of the learners. We have found that there are great differences between the skills of the speakers of the "low-level" subcorpus, maybe because the classification of the database comprises only two skill levels: low and high. The system, as expected, discerns better the incorrect pronunciations of the lowest level speakers.

Comparing the results with those obtained in [127], we have achieved smaller SAs for the phones corresponding to those phonemes not existing in the L1 of the speaker, but better results for phones corresponding to those phonemes that the speaker already knows from her/his native language. As a starting point, we can consider that the results validate the followed strategy to obtain the thresholds for the decisions.

For further details about this work see [137] and [138].

6.3.2 Word-level tests

Word-level scores were also analysed in these preliminary tests. The purpose was to see whether word-level scores are enough —or at least, useful— for Word-by-Word Sentence Verification (WWSV), or phone-level scores have to be considered as well.

At word level, the overall Phoneme Score (PS) for a word can be defined as the weighted sum of the GOPs of its composing phones, as shown in equation (6.6).

$$PS\left(word\right) = \sum_{u=1}^{N} w_u \cdot GOP\left(y_u\right) \tag{6.6}$$

where w_u is the weight of the u-th phone among the N phones composing the word. Typically, the weights are equal for all the phones [139].

Word-level thresholds were calculated using the same methodology as for phonemes (obtaining the incorrectly pronounced words' GOP distribution with simulated errors and computing an EER). Thus, three tests were carried out to check the validity of the thresholds calculated like this.

- **Test 1**: consists in testing the system with 2 218 uttered sentences from the *test* part of the *Basque Speecon-like* database, which include 7 296 words. In this experiment, all transcriptions are correct; therefore, the words labelled as correct by the system are CA and the ones labelled as incorrect are FR.
- **Test 2**: consists in testing the 1 174 isolated words from the *test* part of the *Basque Speecon-like* database, evaluating each file using a randomly selected word of the dictionary, different from the correct one. So, in this experiment the words labelled as correct are FA and the ones labelled as incorrect are CR.
- **Test 3**: consists in testing speech that contains correctly and incorrectly uttered words. For that purpose, 886 sentences from the *test* part of the *Basque Speecon-like* database were used, whose transcription files were modified by deleting one word in each file, thus simulating an erroneously added word. As the original input transcription files contain 5 080 words, 886 uttered words are incorrect, while the remaining 4 194 are correct.

The results obtained for the three experiments are shown in Table 6.4. Despite the good prospects given by the previously calculated scores distribution, in Test 3 the CA *recall* is 99.12 % and the CR *recall* is 84.88 % (CA *recall* can be interpreted as the percentage of correctly accepted words among all the correct words and CR *recall* as the percentage of correctly rejected words among all the incorrect words). This means that incorrect words are not as well classified as correct words are, an asymmetry contrary to what was expected as a consequence of calculating the threshold using the EER.

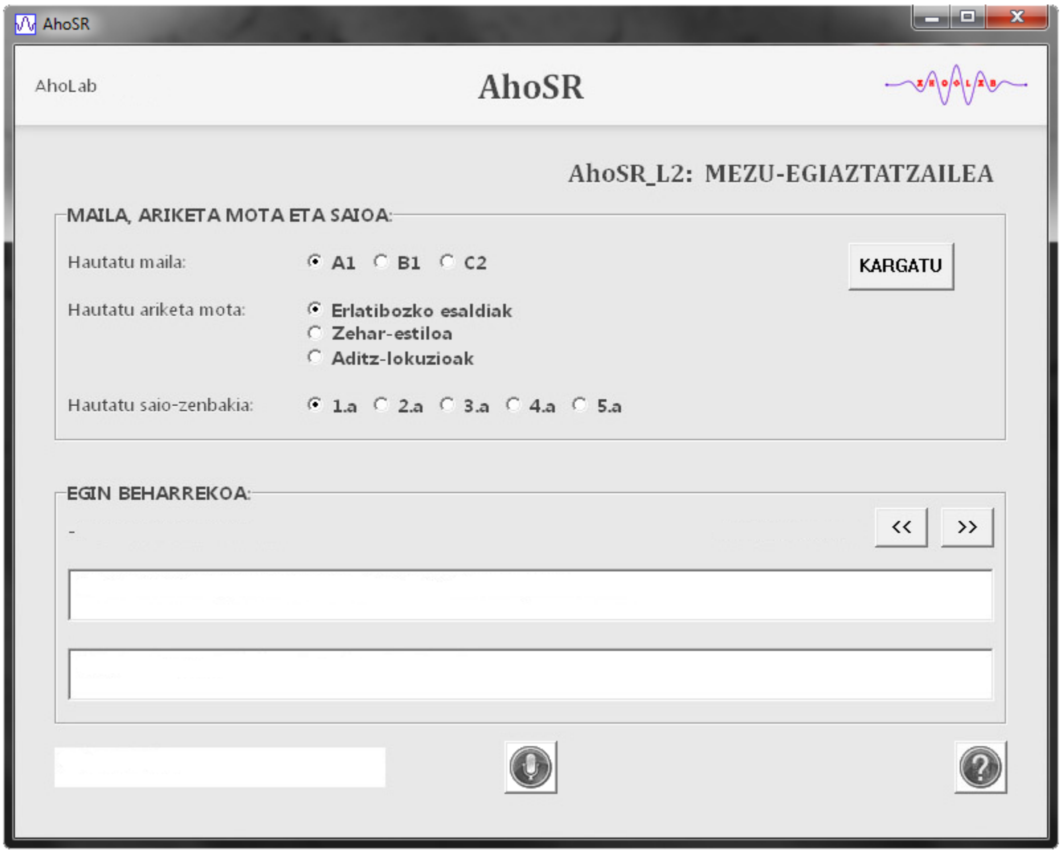
Table 6.4: Results of word-level preliminary tests 1, 2 and 3.

	Test 1	Test 2	Test 3
CA	7 090	—	4 157
CR	—	1 174	752
FA	—	0	134
FR	206	—	37
SA (%)	97.18	100.00	96.63

These preliminary experiments pointed to the need to assess the system in a more realistic environment to obtain a more reliable evaluation. For more detailed information about these preliminary experiments, refer to [96].

6.4 The software

To carry out WWSV experiments in a real environment, a Graphic User Interface (GUI) was built on *AhoSR* (see Figure 6.5). The GUI was designed in C++ for *Windows*, with the idea that ordinary citizens are more used to operate a *Windows* system than any other operating system.



AhoSR_L2: MEZU-EGIAZTATZAILEA

MAILA, ARIKETA MOTA ETA SAIOA:

Hautatu maila: ● A1 ○ B1 ○ C2

Hautatu ariketa mota: ● Erlatibozko esaldiak ○ Zehar-estiloa ○ Aditz-lokuzioak

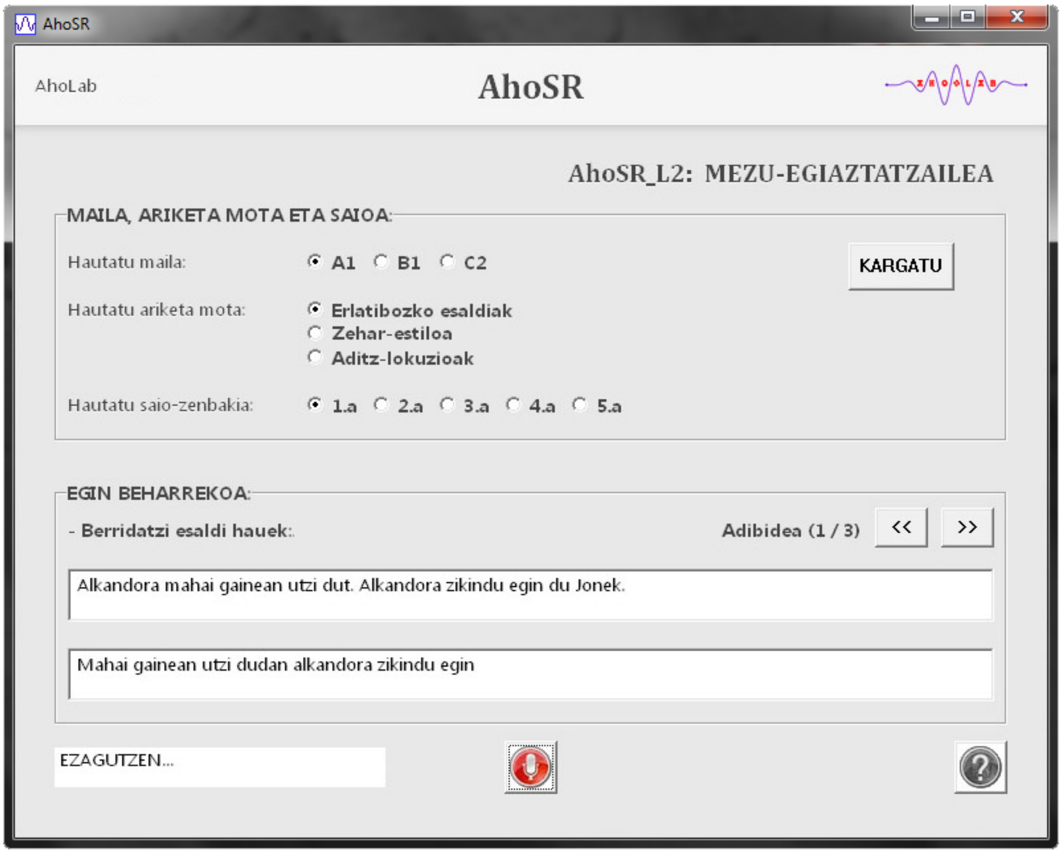
Hautatu saio-zenbakia: ● 1.a ○ 2.a ○ 3.a ○ 4.a ○ 5.a

EGIN BEHARREKOA:

-

Figure 6.5: The AhoSR_L2 system.

The GUI is made up of two main blocks: in the first block, the user can select between different levels of competence (A1, B1 and C2; see [140]), and has the option of choosing the type of grammar exercise, as well as the exercise number. In the second block, the exercise is presented and also a microphone button to start solving the exercise. When the user clicks on it, the button turns red meaning that the decoder has started picking up audio from the microphone. Each time the system verifies a word that the user is expected to utter, it is displayed in a box, so that the user can know on-the-fly if she/he is correctly solving the exercise. Figure 6.6 shows an example of this moment, where the system has verified and displayed several words in the box below. If the user succeeds to the end or a predefined time (20 s) is over, the audio input is stopped and the microphone button turns off again. If the users has not reached the end of the exercise, a "?" button is available to see the rest of the solution.



AhoSR_L2: MEZU-EGIAZTATZAILEA

MAILA, ARIKETA MOTA ETA SAIOA:

Hautatu maila: ● A1 ○ B1 ○ C2

Hautatu ariketa mota: ● Erlatibozko esaldiak ○ Zehar-estiloa ○ Aditz-lokuzioak

Hautatu saio-zenbakia: ● 1.a ○ 2.a ○ 3.a ○ 4.a ○ 5.a

EGIN BEHARREKOA:

- Berridatzi esaldi hauek: Adibidea (1 / 3)

Alkandora mahai gainean utzi dut. Alkandora zikindu egin du Jonek.

Mahai gainean utzi dudan alkandora zikindu egin

EZAGUTZEN...

Figure 6.6: The AhoSR_L2 system running.

It is worth pointing out that if an uttered word does not appear on the screen, either the user has said an incorrect word (CR), or the word has not been correctly verified (FR).

6.5 Exercises and evaluation design

To assess the system in a real L2 acquisition environment, a set of grammar and syntax exercises was selected, initially for A2 level of proficiency in Basque (elementary level). Different exercise types were devised, with the collaboration of three Basque teachers. The exercises were carefully designed to work on grammar concepts requiring a strict word order, such as sentence transformations using relative clauses, reported/indirect speech, or verb expressions. A total amount of 300 different sentences was used for this evaluation.

The evaluation of the tool was performed in two different Basque teaching institutions. A total of 20 volunteers (10 males and 10 females) were recruited for the task. All of them were at the A2 elementary level, and had learned the target material at the classroom. All the volunteers' mother tongue was Spanish except for one of them, whose mother tongue was Catalan. 6 students were between 20-29 years old, 12 were between 30-39 years old and 2 between 40-49 years old. The main characteristics of the volunteers are shown in Table 6.5.

Table 6.5: Involved students' characteristics.

Characteristic		
Gender	Male	10
	Female	10
Age	20-29	6
	30-39	12
	40-49	2
L1	Spanish	19
	Catalan	1

Every student had to solve 30 exercises, divided in 3 different blocks, each block corresponding to a different type of exercise. The students were given the option of practising with three examples before starting. While solving these 3 examples they could ask any question about the system or the type of exercise, but they were not allowed to ask questions while solving the exercises. The tests were done using the same laptop for all the 20 volunteers. They also used the same USB headset with a built-in microphone.

After the test, the students were asked to answer a short questionnaire, in order to obtain a general feedback about their experience. The questions were about the following issues, which were scored from 0 to 10:

- Question 1: The performance of the system.
- Question 2: The usefulness of the system.
- Question 3: The user-friendliness of the system.
- Question 4: The recommendation of using the system in Language Schools.
- Question 5: The overall impression.

6.6 Results

Although the total amount of exercises solved in the evaluation process was 600, only 597 audio files were saved by the tool, because 3 users skipped an exercise inadvertently. After the test, all the recorded files were manually labelled and evaluated with the utterance verification tool. The total Scoring Accuracy and the values of CA, CR, FA and FR obtained are shown in Table 6.6. A total of 2 952 words were evaluated. The total amount of the words uttered by students was 4 404. This figure is not exactly the sum of CA, CR, FA and FR (4 402), as could be expected, because there are two cases where two (incorrectly) verified words come from the same uttered word.

Table 6.6: Word-level Scoring Accuracy of the WWSV experiment carried out in a real environment.

CA	2 419
CR	1 538
FA	136
FR	309
SA (%)	89.89

The Scoring Accuracy obtained in this evaluation performed in a real environment is, as expected, lower than the ones obtained in the preliminary tests (see section 6.3). When the verification tool is used in real life, users make mistakes and the number of rejections increases heavily compared to the lab experiments. This fact demonstrates the utility of the tool and the necessity of working in a real environment to evaluate this kind of applications.

The *recall* and *precision* of CA and CR are shown in Table 6.7. As far as these data are concerned, we deduce that the system behaves in a similar way in both cases, because both *recall* values are comparable. Regarding to *precision*, which can be understood as the percentage of correctly accepted words among all the accepted words (CA *precision*) and the percentage of correctly rejected words among all the rejected words (CR *precision*), we can see a small asymmetry between them. Indeed, the system performance is much better concerning accepted words.

Table 6.7: *Recall* and *Precision* of CA and CR of the WWSV experiment carried out in a real environment.

	CA	CR
Recall (%)	88.67	91.88
Precision (%)	94.68	83.27

Analysing the errors made by the system (FA plus FR), there are 235 files containing errors, out of a total of 597. The distribution of the amount of errors per file (or sentence) is shown in Figure 6.7. As can be seen in the figure, the maximum number of FA per file is 2, and the maximum number of FR is 9. During the test, when the students uttered a correct word but the system was not able to verify and display it, students repeated that word again and again. This happens mostly when verifying short words, even though a frame normalisation factor applies to the confidence score. We noticed that specific phonemes —–such as approximants—– were frequent among the errors. We detected that, because of the FR errors, the system was not able to recover in 17,45 % of the cases, out of the total files with errors.

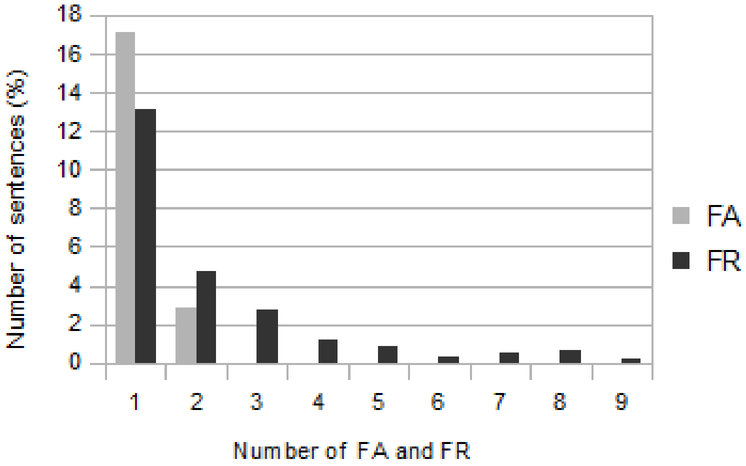


Figure 6.7: FA and FR amount distribution among files.

Concerning the questionnaire that users filled in, results are shown in Table 6.8. Ease of use was scored highest. Although the overall impression of the tool achieves a score of 8.28, the performance has obtained a score of 7, the lowest score. This means that although the system would be recommended (about 90 % score), the users' perception of its performance is considerably lower. It is evident that users weight system errors and hits differently, since, by default, they expect the system to perform correctly.

Table 6.8: Average scores obtained in the students' questionnaire.

Question number	1	2	3	4	5
Average score	7.00	8.25	9.15	8.95	8.28

6.7 Conclusions

In this chapter, the initial experiments of the CAPT and WWSV systems have been described. The CAPT system aims at obtaining a score for each uttered phone related to its corresponding phoneme. The WWSV system is a method to solve grammar exercises (SGP, Spoken Grammar Practice) on the fly, so that the users get feedback of the uttered content instantly. This allows users changing their minds in the middle of a sentence if —for example— the word order they have uttered is incorrect.

GOP thresholds have to be set using the GOP distributions of both correctly and incorrectly pronounced phonemes. However, there is not any pronunciation verification database for Basque, and standard ASR databases lack of incorrectly pronounced data. Thus, a method to calculate the GOPs for incorrectly pronounced phonemes has been devised: the artificially created pronunciation errors. This method consists of inserting isolated changes (or simulated errors) in the dictionary, replacing the phonemes with some other. In order to obtain more conservative results, the phonemes have been replaced with the rest of the phonemes belonging to the same acoustic group. These groups have been obtained clustering GMMs trained for each phoneme, and match almost completely those phonetic groups of SAMPA Basque.

Once thresholds were calculated using the EER of the distributions for the correctly and incorrectly pronounced phonemes, different experiments were carried out at phone-level. The use of native data proved to create better HMMs and GOPs to discern between some problematic phonemes.

Furthermore, word-level experiments were also performed, in order to see how they behave in the WWSV task. Laboratory experiments showed very good results (SA of 97.18 %, 100.00 % and 96.63 % in three tests), but the system needed to be tested in more realistic conditions. Thus, a GUI was build for the system, and a realistic experiment was carried out with 20 Basque students of A2 elementary level. A SA close to 90.00 % was obtained in the experiment (at word-level), a worse result that those obtained in the laboratory, but still encouraging to continue researching in the field and improving the system.

However, this system needs some improvements for a realistic implementation. Nowadays, the trend is to install the speech recogniser in a server to which users can remotely connect. This means that each user will pick up audio with a different device, so that feature normalisation is mandatory. In addition, a VAD is also needed in such a system, in order to get speech segments from a continuous audio signal as well as to save processing time discarding silence frames.

In the next part, we will introduce a new on-line VAD technique (see Chapter 8) and a new on-line CMVN technique (see Chapter 9) based on a novel method: *Multi-Normalisation Scoring*. Besides, the decision making process has been improved using artificial neural networks (see Chapter 10). The result is a web-based CAPT and SGP system, with universal access.

PART III

System improvements

CHAPTER 7

On-line implementation

7.1 Introduction

The goal of the tool introduced here is to be able to be universally accessed wherever the client is located. For that purpose, the most popular solution is implementing a server where all the hard processing is carried out when a client makes a request. As soon as the server finishes its processing, a result is delivered to the client.

Nowadays, the most widely extended client-server system is the web browser, but until very recently, audio-processing on the web has been fairly primitive, and has had to be treated through plugins such as *Flash* and *Java* applets. The introduction of the <*audio*> element in HTML5 has allowed for basic streaming audio playback, and, although being currently under development, it is intended to include more sophisticated capabilities in the future to provide audio mixing, processing and filtering tasks that are found in modern desktop audio production applications. Therefore, the web browser appears to be a good breeding ground for the implementation of this kind of application architectures. The next section (section 7.2) describes in further detail the system architecture for the on-line implementation.

On the other hand, CMVN technique has had to be adapted. In the off-line case, means and variances are computed using all the samples of the whole audio segment being processed. In the on-line case, means and variances have to be initially estimated and then adapted, to ensure that the cepstral features suffer the minimum possible deformation. A study on different strategies to perform this on-line adaptation task is described in section 9.3.

Finally, some concluding remarks are discussed in section 7.3.

7.2 Web technology: HTML5

At the present time, HTML5 along with *Javascript* allows capturing audio from a microphone by means of the Javascript *Web Audio* API [141] and its *MediaStream* interface [142]. Current browser developers are gradually adding HTML5 functionalities and APIs (no browser fulfils the 100 % of the HTML5 requirements yet[1]), and it has taken a long time also for the *Web Audio* API to be widely implemented. However, since 2014 it is supported by almost all the most popular browsers.

The primary paradigm of the *Web Audio* API is of an audio routing graph, where a number of *AudioNode* objects are connected together to define the overall audio rendering. In this context, the *Mediastream* interface manages and captures a stream of media content (audio- or video-related data) from local multimedia devices, such as microphones or video cameras, leaving it ready to upload to the server where it will be processed.

In the Computer Assisted Pronunciation Training (CAPT) task, the recorded audio is not submitted until the user clicks the "send" button. This allows the user to check the recorded data before sending it. So if the user considers that the recorded sentence is not properly uttered or it has any kind of error, audio data can be discarded and rerecorded. A different scenario is needed in the Word-by-Word Sentence Verification (WWSV) task. Audio data must be sent to the server as soon as blocks of samples arrive from the recording device. In this way, the server can detect whether the user has uttered the correct word and, if so, send a feedback instantly, while the audio capturing and sending process continues in the browser. A bidirectional communication must therefore be established.

7.2.1 CAPT system architecture

In this task, the user does not expect real-time response. Since the user will be asked to read a sentence, it is much more convenient to keep the data in the browser so that he or she can check it, once it is recorded. Thus, there is the option to record it again if the sentence has not been properly uttered or even if an external noise has been crept into the audio signal. When the user considers that the recording is of sufficient quality, it can be submitted clicking the *SEND* button.

The audio data can be saved in the browser as a *wav* file. Thus, a simple *php* script in the server can read its header and do the pertinent modifications. Indeed, the HMMs of this study are trained using 16 kHz audio signals, so a downsampling is generally needed. Then *AhoSR* is executed in *WAV* mode, passing as audio input the newly modified audio file. When a result is obtained, it is sent back to the browser using AJAX (Asynchronous JavaScript and XML), which allows updating parts of a web page without reloading the whole page.

In addition, the audio files can be stored in the server, for future research work.

7.2.2 WWSV system architecture

In order to submit the audio data by blocks of samples from the browser to the server, another important HTML5 feature has been used: the *Web Sockets* technology [143]. This is a bidirectional communication technology for web applications which operates

[1] https://html5test.com/results/desktop.html

over a single socket and is exposed via a *Javascript* interface in HTML5 compliant browsers. A *websocket* is not a traditional socket; in practice it combines the parts of UDP and TCP: it's message-based like UDP, but it's reliable like TCP. This feature allows browsers to open a pseudo-connection with a server, exchange data, and close it when the communication is finished. This is very appropriate for sending audio data to the server in an organised and effective way, since HTTP is a stateless protocol, i.e., bearing no information on previous connections. *Websockets* are widely supported by browsers.

In addition, multiple and concurrent connections must be managed by the server using *websockets*. A lightweight and efficient environment for developing server-side web applications managing *websockets* is *Node.js*. This is an open-source, cross-platform runtime environment which contains a built-in library to allow applications to act as a web server. *Node.js* provides an event-driven architecture and a non-blocking I/O API designed to optimise application's throughput and scalability for real-time web applications. Applications are written in *JavaScript* and can be run, within the *Node.js* runtime, on every Operative Systems. Its work is hosted and supported by the *Node.js Foundation*[1], a collaborative project at *Linux Foundation*[2].

In view of the above, the architecture of the WWSV system is composed of three differentiated blocks:

- The browser: where the user interface is located. A *websocket* client is implemented there, which requests a connection with the remote *websocket* server (in the *Node.js* server). When the connection is set, audio from the microphone is captured and processed (downsampled to 16 kHz) on-the-fly by the *Javascript* script located in the browser, and then streamed to the server. Eventual server feedbacks are received as well.
- The *Node.js* server: It is the intermediary between the browser and *AhoSR*. It acts as *websocket* server with the browser, but also as *socket* client to set a connection with *AhoSR*. When a connection request arrives from the browser, the *Node.js* server initiates *AhoSR*, and requests a *socket* connection. If there is no error establishing the *socket*, it also sets the *websocket* connection with the browser. When it receives a *READY* notification from *AhoSR*, it forwards the message to the browser, for the HTML recorder to be displayed.
- *AhoSR*: It must be configured to run in *socket* mode, where audio sample blocks are obtained from a socket connection. Being the *socket* set, as soon as the first data block arrives, it starts the word verification process, and whenever a word is detected, it is instantly sent to the *socket* client (in the *Node.js* server). When

[1] https://nodejs.org/en/foundation/

[2] http://collabprojects.linuxfoundation.org/

the last word is detected or it times out, *AhoSR* sends a *FINISH* message to the client, which is forwarded to the browser in order to stop the audio capturing.

The communication protocol between the three blocks is depicted in Figure 7.1. Note that green text corresponds to the communication between the *websocket* client and the server; blue text to the *socket* client and server communication; and red text means user's actions.

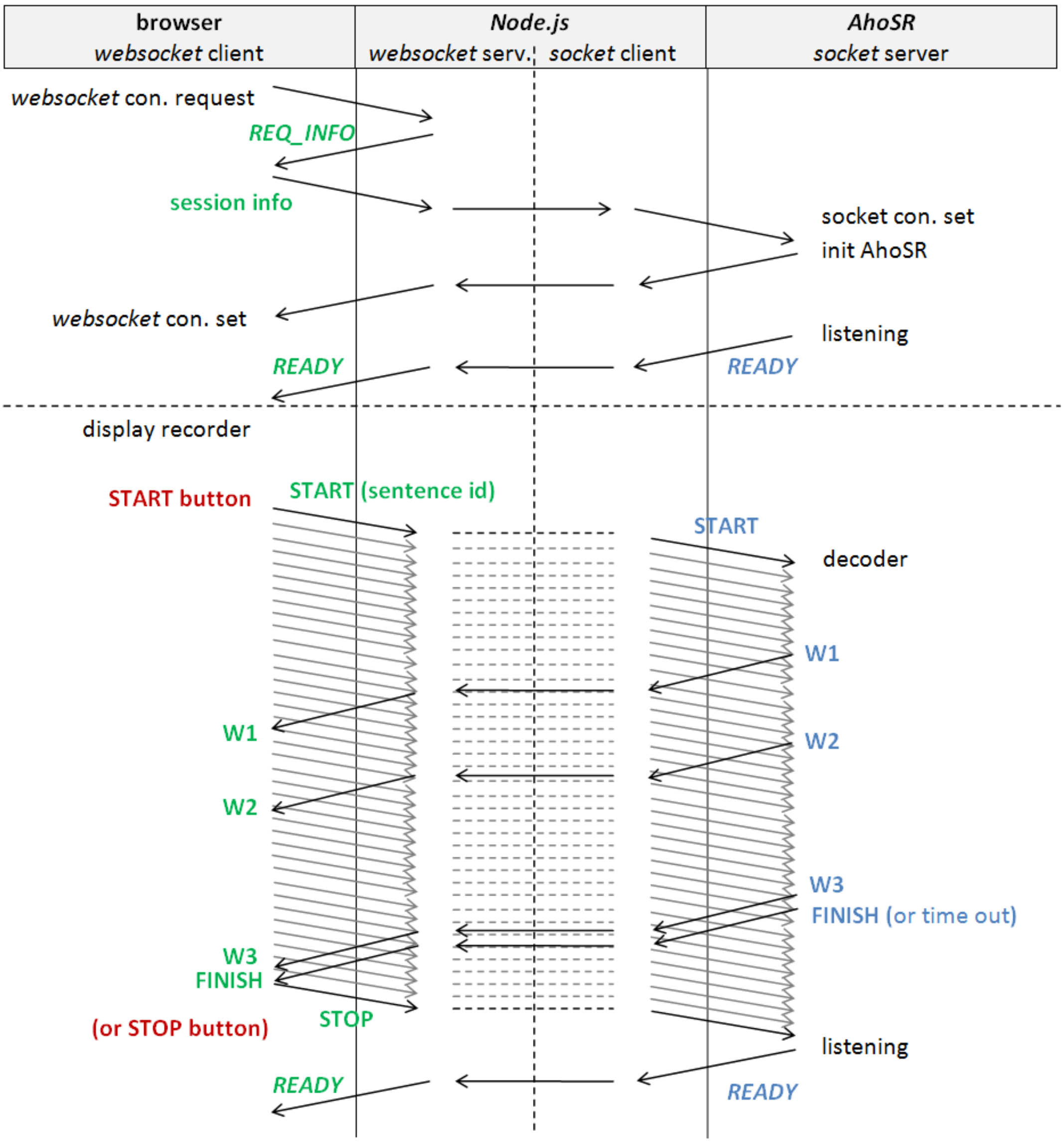


Figure 7.1: Communication protocol along time (y-axis) of the three component parts of the WWSV system: the browser (left), the *Node.js* server (centre), and *AhoSR* (right).

Since each computer records at a different sampling rate, audio samples must be processed. Up to now, the *Mediastream* interface does not allow the sampling rate to be changed, so a function has been expressly added to the audio context in Javascript. Thus, the browser downsamples the audio signals to 16 kHz and sends them directly to *AhoSR*, without any additional processing in the *Node.js* server. If this downsampling were not performed, it would not be possible for *AhoSR* to obtain the sampling rate of the signal.

A key determining factor for proper functioning of the system is the size of the data buffer where the incoming audio samples are set. When the data buffer is full, an event is called and the whole block can be sent at this moment. This means that it will affect the resolution of the system. The size must be one of the following values: 256, 512, 1024, 2048, 4096, 8192, 16 384. Lower values will result in a lower latency, since smaller sample blocks are sent more frequently. However, depending on the capacity of the user's computer, audio breakups and glitches can occur. After some testing, it has been concluded that a good compromise is 1024 or 2048.

The audio samples can be saved in a *wav* file both in the *Node.js* server and in *AhoSR*, in an effort to harness material for future work.

7.3 Conclusions

This chapter explains the adaptations needed to implement our system in a server, universally accessible. On the one hand, a combination of two APIs belonging to the recent HTML5 specifications have been put working together. The *audio API* allows the browser to record data from a microphone, so that it can be sent to the server. For CAPT, the audio can be sent as an entire file after the user has checked it; for WWSV, we make use of the *websocket API*, which allows setting a socket-like connection between the browser and the server. Thus, the audio collected through the microphone can be sent as soon as it is recorded, and feedback received.

CHAPTER 8

On-line VAD

8.1 Introduction

Voice Activity Detection (VAD) is an important issue in ASR or ASR-based systems. Using VAD, audio signals are split into autonomous speech segments before being passed to the subsequent modules. By only passing speech frames, the computation cost of the recogniser reduces and, as a consequence, the response time of the decoding process [144]. For systems that, as the one introduced in this thesis, the access 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 systems is to cope with background noise in the input speech signal [145].

Two kinds of errors must be considered: silence or noise segments being passed as speech (the *non-speech error rate*) and speech segments being misclassified as silences and then not being passed to the processing system (the *speech error rate*). Both must be kept low of course, but their importance depends on the needs and design of the system using the VAD.

VAD is typically the first module employed in acoustic processing systems. It is profusely used in the development of all kinds of expert systems. In [146] the authors use ASR technology with a VAD to develop a dialogue system in a motorcycle environment. [147] describe an integrated system for processing voice emergency commands using a VAD followed by ASR. VAD and ASR technologies constitute the core of the speech interface in a system using a serious game to support therapy for mental disorders in [148]. All these systems, based on ASR, require a very low ratio of lost speech frames in order for all the meaningful audio frames to be available to the recogniser. On the other hand, if non-speech segments are passed as speech the recogniser will still be able to detect them, as they typically have a silence (or non-speech) model. The main purpose of VAD in ASR interfaces is to eliminate long silences and split the audio stream into shorter, manageable segments. Additionally computation time is reduced and consequently so is the decoding response time.

ASR is not the only technology that requires a good VAD module. [149] identify VAD as one of the research areas for speaker recognition. For instance, VAD is included in an intelligent porch system where people are identified by their voices before entering the house [150]. VADs are also an important module in speaker segmentation and clustering systems, such as the diarisation system presented in [151]. In addition, VAD is an essential module in systems that include emotion identification [152]. For speaker and emotion recognition systems the VAD employed requires a very low number of erroneously classified silence or noise frames, since silences or noise frames do not convey emotion or the speaker's identity. A high non-speech error rate will thus lower the performance of the system. If however some speech frames are lost, the system will still be able to perform correctly.

Current VADs can be tuned to behave closer to one mode or to the other, though the ideal behaviour would of course be to reduce both non-speech and speech error rates as far as possible.

When the oral interface of a system picks up audio signals by means of different devices and in different environments, the VAD has to cope with different recording conditions, channel characteristics and noise levels. This is in fact the greatest challenge for the current ASR systems [145]. VAD systems currently adapt different parameters to adjust to changing background noise conditions. However, this approach has its shortcomings: on the one hand, there is a need for an initialisation time over a segment to adjust the parameters, which introduces an undesirable delay. On the other hand, any incorrect estimation of the parameters will lead to uncertainty in the performance of the system [153]. Training the VAD beforehand is one way to avoid the initial adaptation, but the trained system should be able to generalise to unseen channels or background noises. On-line VAD decision making is still a challenge.

From the point of view of acoustic features, very different parameters have been investigated: periodicity measure [154, 155], zero-crossing rate [156], pitch [157], Short Term Energy (STE) [158] and Long Term Energy (LTE) [159, 160], spectrum analysis [161, 162], cepstral distance [163], Linear Predictive Coding (LPC) [164] and combinations of different features [165]. More recent research has been focused on using multiple features to train a statistical model or classifier using machine learning techniques rather than on exploring more discriminative new acoustic features, which was the traditional trend.

Both Gaussian Mixture Models (GMM) and Hidden Markov Models (HMMs) have been tested in the context of VAD. In [166], speech and non-speech segments are modelled by two HMMs. A simple grammar is used to model transitions from one HMM to the other and voice detection becomes a task of finding the best path through a recognition network. It is shown that a simple HMM-based VAD functions properly when clean signals are considered. In [167] the same HMM strategy is followed to deal with background noise, but acoustic features and normalisation operations are used along with the results conveyed by the HMMs. In [168] several noisy HMMs are trained to detect different noisy non-speech segments. In this thesis we also use the approach of scores generated by the HMMs.

[169] addresses the problem of far-field speaker interference in human-machine oral interaction. A decision tree (DT) is trained using the scores of speech/non-speech HMMs and additional information related to far-field speech. A Support Vector Machine (SVM) is used in [170] to discriminate between speech and non-speech, and improved versions include Signal to Noise Ratio (SNR) information as in [171, 172]. Hybrid SVM/HMM architectures are also proposed for VAD in [173] to retain the discriminative and non-linear properties of SVM while modelling the inter-frame correlation through a HMM. Results show a better performance for the SVM-based VAD system. However, relatively high speech error rates are still obtained. Our proposed VAD outperforms this technique and obtains a speech error rate more than three times lower.

More recently, neural networks (NN) have appeared in the literature of VAD approaches. For instance, [174] uses a recurrent neural network (RNN) with perceptual linear prediction (PLP) features testing clean signals. Convolutional neural networks (CNN) are also used in [175] with mel-spectral coefficients, but adaptation with supervised data is needed for unseen channels. In [176] feature vectors consisting of log-mel filterbank energies are fed into a DT, an SVM and a CNN classifier. However, in this VAD approach several parameters must be adjusted to adapt to different noise conditions.

Regarding on-line performance, the current deep learning approaches tend to have very long inference times, mainly because neural network architectures are normally designed to be as complex as possible without considering real-time limitations [177]. An exception is the system introduced in [178], where a collection of different acoustic features are used to train a deep-belief neural network (DBNN). Extensive experimental results where different types of noise are tested show that it outperforms several reference VADs, even in real time. Nevertheless, this system has to compute almost 300 features in each frame, which increases system complexity. By contrast, our approach seems to get better results and is much simpler.

In this chapter we present a simple but highly effective VAD based on a method that we have called *Multi-Normalisation Scoring* (MNS). This consists of classifying multiple observation likelihoods generated by an HMM trained with normalised Mel-Frequency Cepstral Coefficients (MFCC) corresponding to silence audio segments. Our proposed VAD technique makes use of a classifier which is trained beforehand, so that only a classification task needs to be performed when a new incoming speech frame arrives. This means that results are obtained on-line frame by frame and there is no need to adjust any parameter, so no initialisation period is needed. Furthermore, in comparison with two current standard ITU-T VAD algorithms, our VAD has proved to perform much better in labelling non-speech frames and to obtain similar results in labelling speech frames without increasing computing time. The VAD has been tested for different types of noise, as well as several SNR. The results show that our proposed VAD technique is able to generalise. This research has been published in [97].

The chapter is organised as follows: Section 8.2 is a study of different aspects of the observation likelihood scores obtained by a silence GMM. A preliminary experiment is also described, which shows that these scores are useful to distinguish between silence and non-silence segments. section 8.3 describes the general architecture of the VAD system proposed in this paper. section 8.4 describes the MNS method and its motivation. section 8.5 provides a short overview of the databases used. To assess the performance of the new VAD, several databases have been chosen in an attempt to cover a variety of contexts and use a considerable amount of test speech material. The results of different experiments (under both clean and noisy conditions) are shown in section 8.6. section 8.7 describes a validation experiment comparing the results with two standard VADs, and some conclusions are finally drawn in section 8.8.

8.2 Observation likelihood

In speech recognition, audio segments corresponding to the same recognition unit (word, phone, triphone etc.) are gathered and processed in order to extract acoustic features (typically MFCCs) from them and train a different acoustic model for each unit. HMM is a very popular acoustic model, since it not only models the likelihood of a new observation vector but also the sequentiality of the observations.

Observation likelihoods are generated by GMMs, each of which corresponds to an HMM state. For an observation vector o_t, the observation likelihood b_j of a GMM at the j_{th} state is calculated as shown in equation (8.1).

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} N\left(o_t; \mu_{jm}, \Sigma_{jm}\right) \tag{8.1}$$

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 μ and covariance matrix Σ.

In this work, the observation likelihoods have been obtained from the silence HMM trained using the *Basque Speecon-like* database [101], specifically the *close-talk* channel.

8.2.1 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 13 second order derivatives as acoustic parameters, and 32-mixtures 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 $y = \{y_1, y_2, ..., y_N\}$, their mean μ_N and variance σ_N^2 vectors are calculated as defined in equation (8.2) and equation (8.3), respectively.

$$\mu_N(i) = \frac{1}{N}\sum_{n=1}^{N} y_n(i) \tag{8.2}$$

$$\sigma_N^2(i) = \frac{1}{N}\sum_{n=1}^{N} \left(y_n(i) - \mu_N(i)\right)^2 \tag{8.3}$$

where i is the i^{th} component of the vector.

The cepstral features are then normalised using the calculated mean and variance vectors, as given in equation (8.4). Thus, each normalised feature has zero mean and unit variance.

$$\hat{y}_n(i) = \frac{y_n(i) - \mu_N(i)}{\sigma_N(i)} \tag{8.4}$$

8.2.2 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 weather CMVN is applied or not. Figure 8.1 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 normalisation respectively, through a utterance composed of four words. In this case, the normalisation has been performed using the means and variances computed from the file.

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.

8.2.3 The central state of a silence HMM

The central-state in a three-state HMM is a priori the most stable state of the model, since the states of the ends have to cope with transitions between models. It makes sense to believe that the same will happen to the silence HMM, where states from the end have to model transitions between silence and speech.

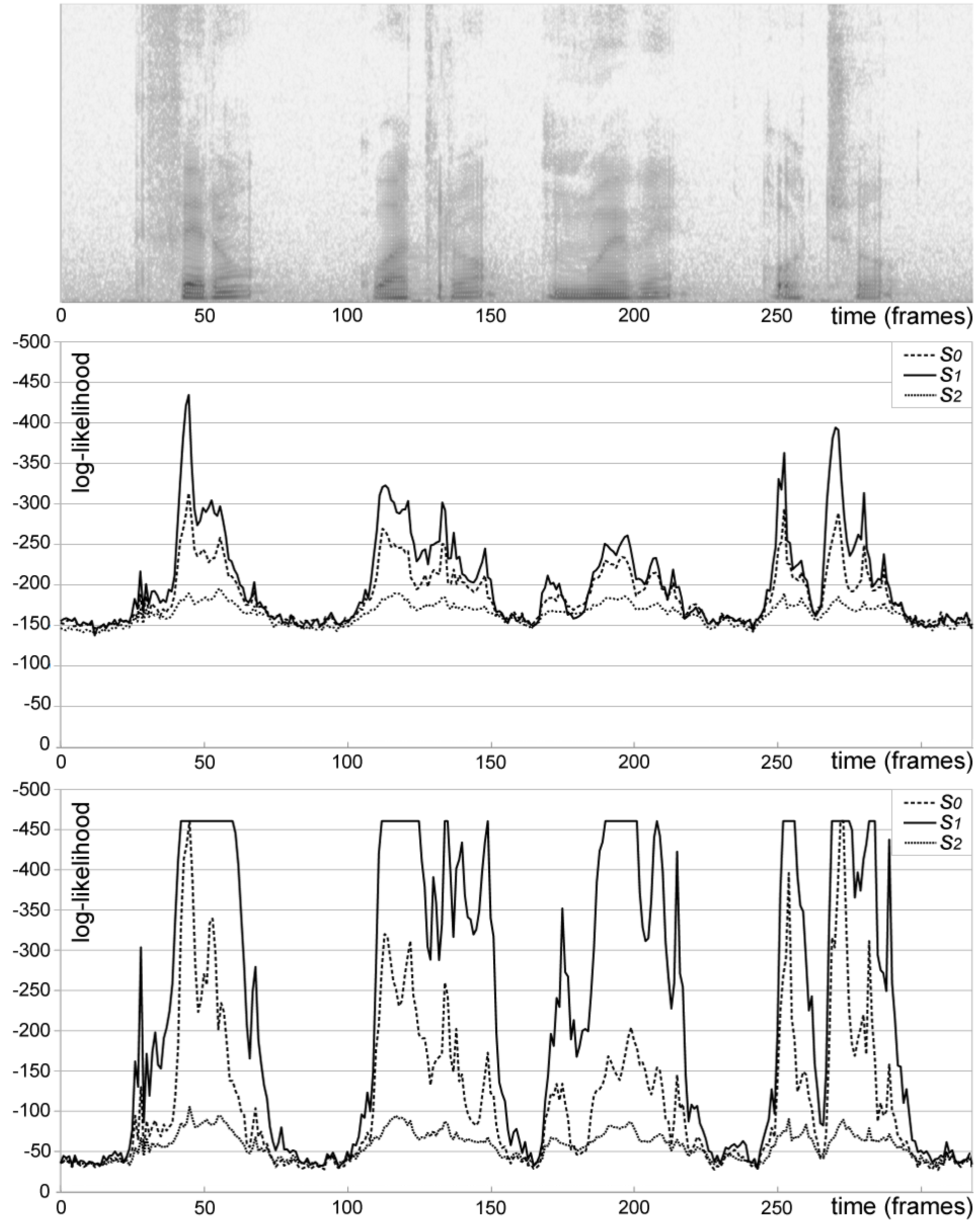


Figure 8.1: Spectrogram (top) and observation log-likelihoods along time (frames) generated by the left state (s_0), central state (s_1) and right state (s_2) of the silence HMM without CMVN (middle) and applying CMVN (bottom).

An illustrative example is provided in Figure 8.2, which shows the log-likelihoods generated by the GMM of each HMM state (s_0, s_1 and s_2) through an utterance composed of three words (notice the mouth click just before the second word). The observation likelihood curve generated by the central-state (s_1) GMM seems much more discriminative than the ones at the ends, which are more irregular. The analysis was carried out using the silence HMM from the "*M25 — R+M25*" HMM set (see Chapter 5) with offline CMVN (trained using the *close-talk* subset, with 13 MFCCs and 13 first and 13 second order derivatives as acoustic parameters, and 32 mixtures GMMs).

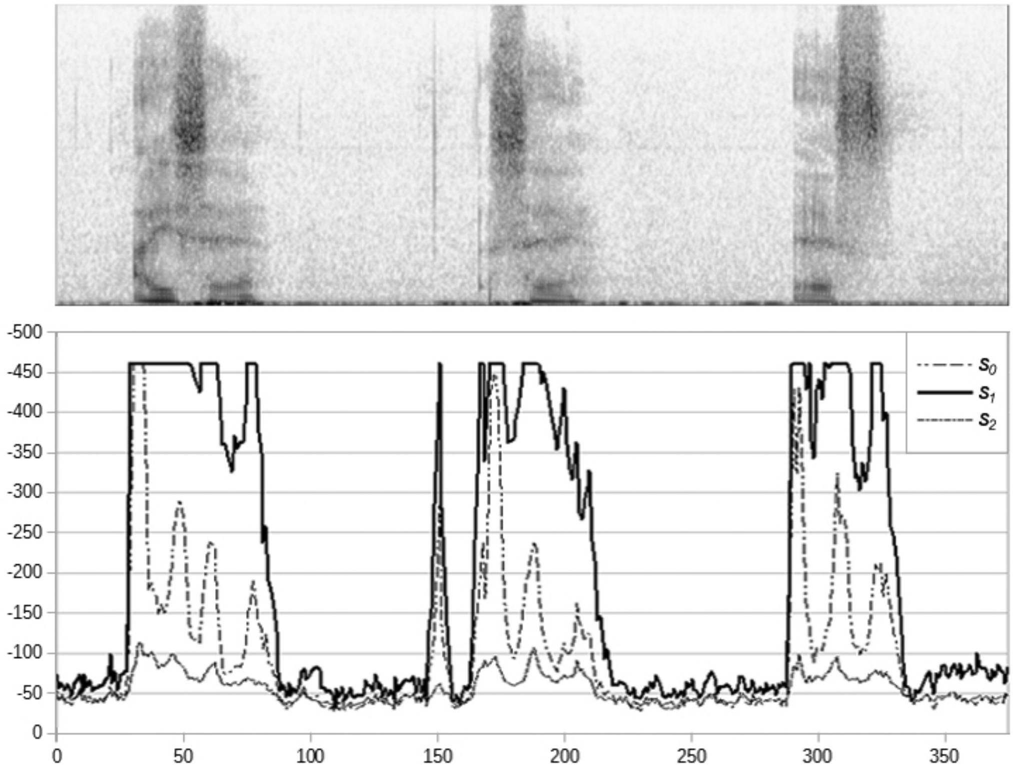


Figure 8.2: Spectrogram of an utterance consisting of three words (top) and observation log-likelihoods over time (frames) generated in the left state (s_0), central state (s_1) and right state (s_2) of the silence HMM trained with normalised MFCCs (bottom).

Initially, we witnessed that the likelihoods generated by the central-state GMMs of silence HMMs behaved very differently depending on the way the HMMs were created. Different HMM sets were tested, resulting that the response significantly degrades as the training data set includes more inaccuracies. The HMM sets which show the best discriminative behaviour —and so are expected to produce the best results— are those created using the $M25$ subset (manually labelled) at the first stages of model training. So the importance of a correctly phased training for our database is in evidence. In Figure 8.3, the log-likelihoods generated by the central states of different HMMs are shown, being the most discriminative the darkest line (*M25 — M25+R*) and the most irregular the lightest line (*M25+R*).

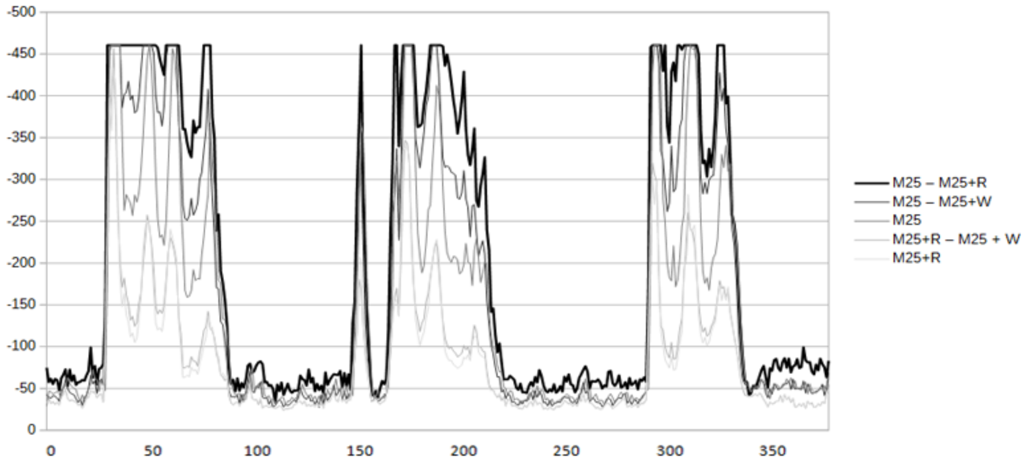


Figure 8.3: Observation log-likelihoods along time (frames) obtained at the central state (s_1) of differently trained silence HMMs using normalised cepstra.

Another interest point to focus on in a voice activity detector is its robustness against noise. Figure 8.4 shows the likelihoods generated by the same utterance under four different noise conditions (obtained from the *Spanish SpeeCon*, see section 8.2.4). The utterance was recorded at four distances using four different microphones. Each recording shows a different channel (C_0, C_1, C_2 and C_3) with different SNR values: around 20 dB the cleanest (C_0), and around 0 dB the noisiest (C_3). The utterance contains the following digit sequence in Spanish: "*cero*, *cuatro*, *nueve*, *ocho*" (zero, four, nine, eight).

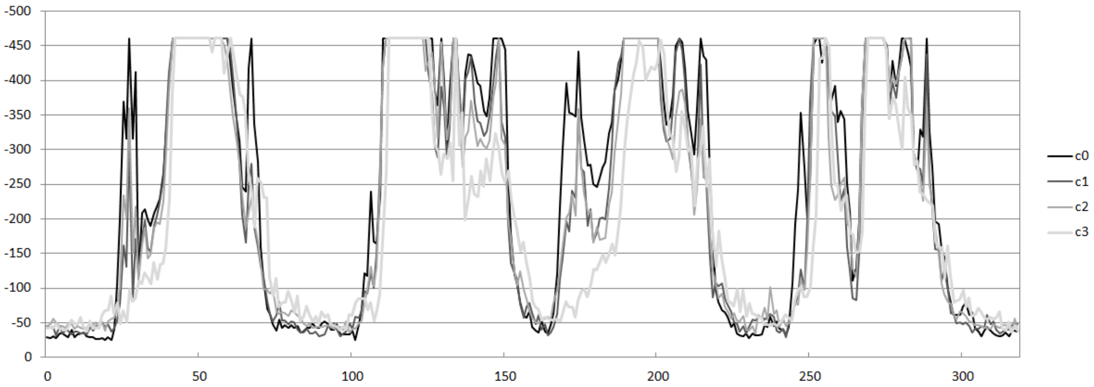


Figure 8.4: Observation log-likelihoods along time (frames) obtained at the central state (s_1) of the silence HMM when processing different SNR audio signals: from C_0 (20 dB) to C_3 (0 dB).

Probability curves show that, as expected, a degradation occurs when processing noisier signals, but even so the curves remain rather discriminative. At 0 dB, the most adverse effect occurs at the initial and ending phones, where, depending on the phone, the probabilities can be very similar to the probabilities of the noisy silence. This happens mostly when the initial phone is a noisy phone. However, the VAD shows a good behaviour under the other less noisy conditions, with probability profiles very similar to those obtained with a SNR of 20 dB.

8.2.4 Preliminary VAD experiment

To assess the stability of the observation probability curves generated by the central state of the silence HMM, a VAD accuracy experiment has been carried out. Each file's features have been normalised using the means and variances calculated from the same file. So we consider it is an off-line VAD experiment, since each file has been pre-processed before starting to classify each frame one by one. The on-line calculation of normalised cepstra involves initial estimation issues (see section 9.3) that would dramatically affect to the behaviour of the VAD. However, the MNS approach in the next section solves this issue successfully.

For the experiment, the most discriminative silence HMM tested in the previous sections has been chosen: the "*M25 — R+M25*" silence HMM. Regarding the tested files, a subset of the *Spanish SpeeCon* database [100] has been chosen, since we have used it in a previous research work as reference to assess different VAD algorithms [179]. This subset contains more than 1000 utterances picked up in different environments (office, entertainment, car and public place). Each utterance was recorded by means of four different microphones: a close-talk 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). Therefore, each of these channels contain signals recorded at different environments, C_0 being the cleanest and C_3 the noisiest scenarios. The signals in the database were recorded at 16 kHz sample rate and 16 bit per sample. The reference speech and silence labels are those used in the mentioned previous work.

The VAD accuracy experiment consists in evaluating the ability of the system to discriminate between speech and silence segments at the four 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^{ref} and N_1^{ref}, respectively), as shown in equation (8.5). In addition, the TER (total error rate) has also been computed as the quotient between the total number of incorrectly classified frames and the number of total frames (equation (8.6)).

$$ER_0 = \frac{N_{0,1}}{N_0^{ref}} \times 100; ER_1 = \frac{N_{1,0}}{N_1^{ref}} \times 100 \tag{8.5}$$

$$TER = \frac{N_{0,1} + N_{1,0}}{N} \times 100 \tag{8.6}$$

The experiment has been performed for different thresholds and different speech and silence segment durations. The best results were obtained setting 15 frames as the minimum duration for both speech and silence segments, and the corresponding results are shown in Figure 8.5. The best TER results have been obtained for $Th = -150$ at the various SNR levels. Besides, a fairly flat segment can be seen between around -150 and -200 in the curves, confirming a stable performance margin. The equal error rate (EER) points of ER_0 and ER_1 curves are located between these two values.

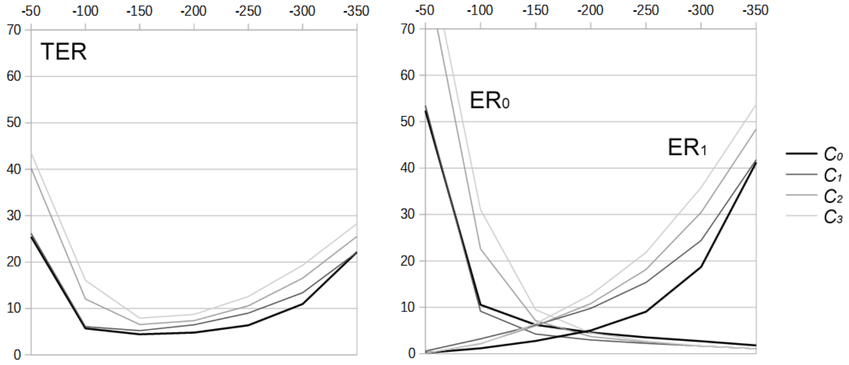


Figure 8.5: Off-line VAD accuracy experiment: TER (left picture) and ER_0 and ER_1 (right picture) for different threshold values at the four SNR levels.

Table 8.1 shows the results obtained for $Th = -150$ at the four SNR levels. The results show, as expected, that the error rates increase as the SNR decreases. However, the best silence error-rate is obtained at C_1.

Table 8.1: Off-line experiment TER, ER_0 and ER_1 for $Th = -150$ and the four different channels.

	TER	ER_0	ER_1
C_0	4.42	6.21	2.74
C_1	5.21	4.22	6.13
C_2	6.53	7.10	6.00
C_3	7.90	9.46	6.45

For speech processing, it is important to reduce the ER_1 as much as possible. To this end, an additional margin of 10 frames has been set to speech segments, since many speech classification errors appear at the boundaries of the segments, mostly in those words beginning with a noisy phoneme. This allows decreasing significantly the ER_1, with a not very significant resulting TER degradation. Note that boundaries are always imprecise. So much so that in some evaluations intervals of ± 5 ms around them are usually discarded from the evaluation process [180]. Table 8.2 shows the results of this other experiment, where the results of the proposed technique are compared with the results of four popular standard VAD algorithms, over the same data-set [179]: the G.729 system [156], the FD (frame-dropping mechanism) and NR (noise reduction system) algorithms implemented in AFE-DSR (advanced front-end for distributed speech recognition) [181] and the LTSE (long-term spectral divergence) algorithm [182].

Table 8.2: Comparison of different VAD algorithm results at four SNR levels.

(c) Total Error Rates (TER)

	G.729	AFE-FD	AFE-NR	LTSE	Prop.
C_0	28.98	30.49	28.11	18.68	7.99
C_1	38.74	26.24	27.73	16.22	7.20
C_2	38.16	25.09	20.69	19.02	8.71
C_3	42.94	24.61	27.05	17.54	9.99

(b) Silence error rates (ER_0)

	G.729	AFE-FD	AFE-NR	LTSE	Prop.
C_0	56.06	63.88	63.88	58.23	15.68
C_1	70.23	54.75	54.75	55.96	12.42
C_2	59.54	59.54	52.10	38.10	15.39
C_3	70.49	70.49	50.10	47.65	17.59

(a) Speech error rates (ER_1)

	G.729	AFE-FD	AFE-NR	LTSE	Prop.
C_0	3.63	0.03	0.62	0.05	0.79
C_1	9.28	0.23	1.98	0.49	2.30
C_2	18.19	0.48	4.83	0.53	2.47
C_3	17.22	1.41	8.30	1.34	2.89

Table 8.2 shows that the best total results are obtained with the proposed technique, taking into account speech and silence together. Regarding the speech, the AFE-FD

and LTSE systems obtain better results, however they show the disadvantage of having very high silence classification error rates for all the channels (the lowest result is 38.10 %). This means that many silence frames would be sent to the recogniser. Our initial system produces error rates between 12.42 and 17.59 % for silence frames classification.

8.2.5 Conclusions

Summing up, results show that the discrimination capacity of the central-state GMM of the silence HMM, despite its simplicity, is very good. The overall error rates obtained are about the half of the second best system tested. However, the most important rate is the error rate of speech frames (ER_1), since silence frames which are let pass to the recogniser can still be classified as silence by the decoder. Our system shows very low error rates for all the channels, being the highest of 2.89 % for the channel with the farthest microphone (C_3).

8.3 General architecture of the MNS-based VAD system

The on-line VAD technique proposed in this thesis consists of three core blocks, as shown in Figure 8.6. The input to the system is a vector of MFCCs obtained from the current signal frame, and the output is a VAD label: *speech* or *non-speech*.

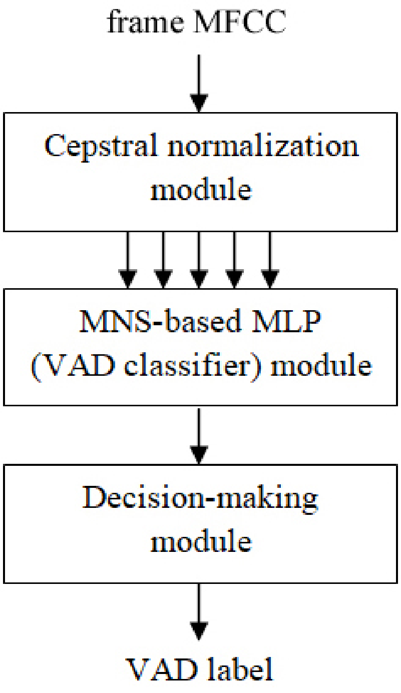

frame MFCC
Cepstral normalization module
MNS-based MLP (VAD classifier) module
Decision-making module
VAD label

Figure 8.6: General architecture of the on-line VAD technique proposed here.

The three core blocks are:

- Cepstral normalisation module: the acoustic features (MFCCs) of the incoming signal frame are normalised using different normalisation factors.
- MNS-based MLP (VAD classifier) module: this module classifies a vector obtained by our proposed MNS method, using a Multi-Layer Perceptron (MLP).
- Decision-making module: this implements a finite-state automaton to make immediate decisions in order to cope with glitches and enhance the results.

Block 2 implements the method presented here, and is described throughout the thesis. Blocks 1 and 3 are described in more detail in the following subsections.

8.3.1 Cepstral normalisation

Cepstral normalisation is essential to develop the VAD proposed in this work. Indeed, as demonstrated in earlier works [183], the observation likelihoods generated by the silence GMM trained with normalised MFCCs follow a fairly discriminative pattern for speech and non-speech frames. The VAD proposed in this work takes advantage of this characteristic.

Overall, parameter normalisation is indispensable to create robust acoustic models and cope with audio signals captured in different environments. The spectral subtraction approach of [184] is well established in the ASR field for compensating for the differences (channels, background noise, etc.) in the incoming signals. However, the most common practice is to perform CMVN on the extracted features, as it outperforms spectral subtraction techniques [185].

As explained in [135], the mean of an MFCC over N frames conveys the spectral characteristics of the current microphone and room acoustics. At the limit, when $N \longrightarrow \infty$ for each utterance, the means from recordings of the same environment can be expected to be the same. Thus, cepstral mean normalisation (CMN) permits the removal of a stationary, linear channel transfer function; and variance normalisation (CVN) helps to compensate for the reduction of the variance of the MFCCs due to additive noise.

The classic CMVN approach [186, 187] seeks to estimate mean and variance vectors per cepstral feature (MFCC). The feature vectors are then shifted and scaled by the estimated means and variances, so that each normalised feature has zero mean and unit variance. An effective solution for calculating reliable means and variances is to estimate them using the whole utterance (*off-line* performance). This utterance-based normalisation can result in undesirable delays, since utterance processing cannot begin until the last frame arrives. In time-synchronous (or *on-line*) systems, windows of a minimum length of 150-200 ms are typically used as a compromise between the quality of the estimated means and variances and the latency. Once an initial value is estimated, some type of recursive normalisation is usually applied.

The initial values for means and variances can be estimated using the first M frames (and then adapting recursively). Correct estimation of these initial values depends heavily on whether these M frames contain speech or not. If there is no speech in them, computed variance values will be very small, which will strongly amplify the amplitude of the normalised signal, and vice versa. In consequence, a good estimation of the initial values for means and variances is of the utmost importance. This issue can be overcome by using the method introduced in this thesis, which is based on applying multiple normalisation factors to cepstral features. This enables decisions to be made frame by frame with no need to use a window.

8.3.2 Decision-making module

As the decision of speech/non-speech is made frame by frame, very short segments labelled as speech can appear in the output of the VAD. These short segments usually correspond to noises and glitches and degrade the performance of the following processing system. In an off-line implementation there is usually post-processing, but on-line implementation means making immediate decisions. In our on-line implementation, a classic state-diagram is implemented (see Figure 8.7). Two parameters are considered: *minimum speech duration* (T_{min_speech}) and *minimum silence duration* (T_{min_sil}), which set the minimum number of frames that a segment must contain to be considered as speech or silence (non-speech), respectively. As can be seen in the figure, if the VAD changes its state from non-speech to speech (or vice versa) in a given frame, the next T_{min_speech} frames (or T_{min_sil}) are also analysed. If the result of checking these frames matches the state of the current frame, a state change is made; otherwise it is assumed that there has been a glitch and the VAD does not change its state. Obviously, this method adds a short delay each time a state change is found, but no delay is added when the same state is maintained. This enables the system to completely recover during non-transitional segments.

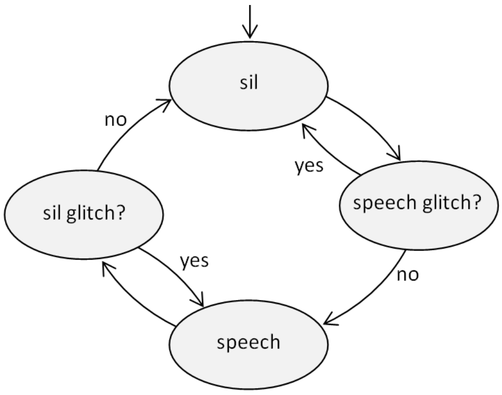


Figure 8.7: State diagram of the on-line implementation of the VAD.

For the experiments carried out in this thesis, a minimum segment duration of 15 frames was empirically chosen for both T_{min_speech} and T_{min_sil}.

8.4 The MNS method

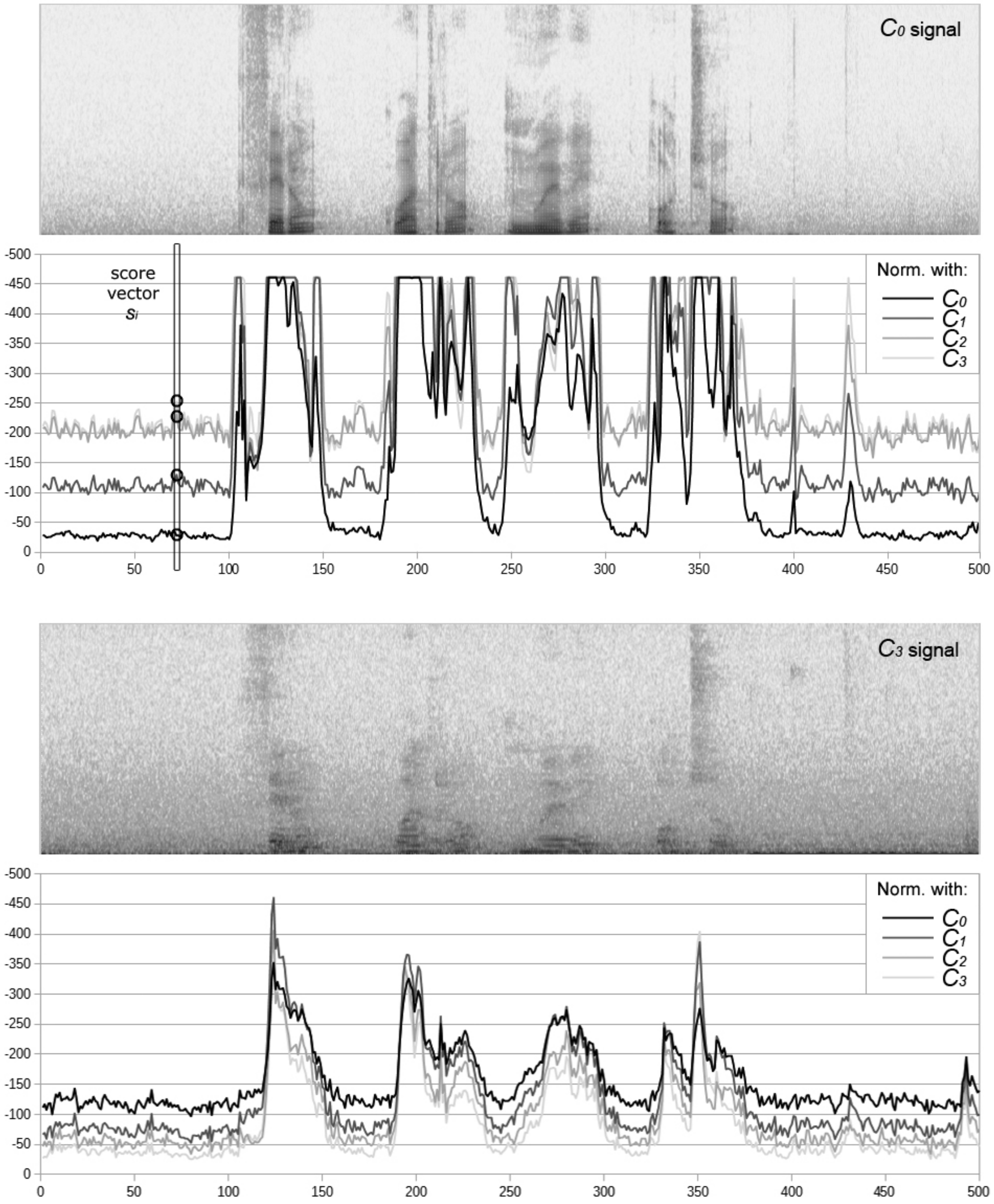


Figure 8.8: Spectrogram and central-state silence HMM observation log-likelihoods of a C_0 signal (top) and an C_3 signal (bottom) over time (frames) for different normalisation modes using pre-calculated means and variances from datasets C_0, C_1, C_2 and C_3. The vertical narrow box marks the vector of scores in frame i.

The MNS method consists of generating multiple observation likelihood scores by normalising the MFCCs using means and variances computed from different speech datasets obtained under different recording conditions. The observation likelihood vectors thus obtained can characterise the behaviour of the speech and non-speech frames in different conditions. As an illustrative example, Figure 8.8 shows the behaviour of the scores obtained by normalising a signal picked up from near (top, C_0) and another from afar (bottom, C_3) with the pre-calculated means and variances obtained from the four datasets recorded simultaneously at four distances: close (C_0), desktop (C_1), medium (C_2) and far (C_3).

Assuming that the differently normalised scores of the non-speech segments follow a pattern (see score vector s_i in Figure 8.8), it is likely that the speech scores do likewise. If so, only a good classifier would be needed to detect those patterns and classify the vector as belonging to a speech or non-speech frame.

8.4.1 The classifier: MLP

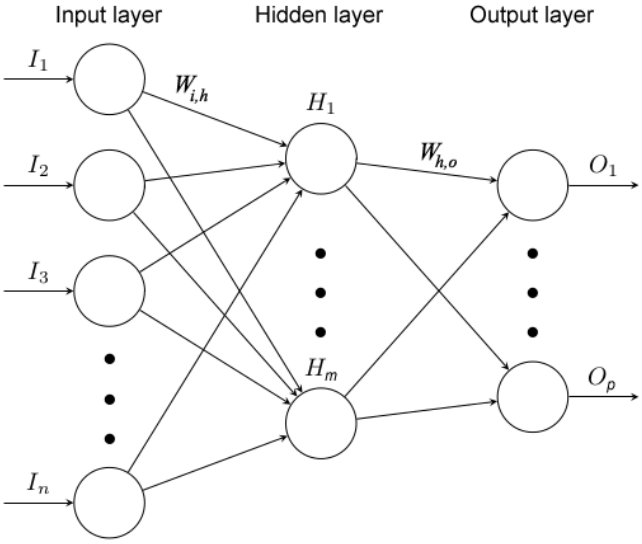


Figure 8.9: Single hidden layer MLP neural network, with n, m and p nodes in the input layer, hidden layer and output layer, respectively.

Different classifiers have been tested in order to see whether the scores obtained from the differently normalised acoustic features (or Multi-Normalisation Scoring, MNS) are valid, and the best results have been obtained using a MLP [188][189]. A MLP is a feed-forward Artificial Neural Network (ANN) model that maps input data onto a set of appropriate outputs. Considering input data as the attributes of a new observation and the outputs as categories, this process can be considered as a classification task.

As shown in Figure 8.9, a MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a non-linear activation function. The training process of a MLP model consists on recursively calculating the weights between nodes by measuring the deviation between the output and the expected answer. This is performed using the back-propagation technique —a supervised learning technique— [190]. Unlike the standard linear perceptron [191][192], it can distinguish data that are not linearly separable [193]. For the next experiments, MLPs were trained using WEKA (Waikato Environment for Knowledge Analysis), a popular free, open-source software written in the Java™ language for data-mining tasks [194, 195].

8.5 Speech databases

In this section, we can see the most important information about the databases used to carry out the MNS-based VAD experiments. In total, four speech databases have been used. Firstly, we used the *Basque Speecon-like* database [101], specifically the *close-talk* channel, to train the HMM for silence frames. Using this HMM, an MLP was trained by applying the MNS method to the files of the *Basque Speecon-like* database and a subset of the *Spanish Speecon* database used in an ECESS evaluation campaign of voice activity and voicing detection [180].

The first VAD experiment was performed by testing the files from a third database: the *TIMIT Acoustic-Phonetic Continuous Speech Corpus* [196]. The second VAD experiment was carried out by testing the system with noisy signals. For that purpose, the *Noisy TIMIT* speech database [197] was considered, in particular the *babble* noise dataset and *white* noise dataset *Test* blocks. Each dataset comprises 10 subsets each of which corresponds to a different SNR (from 50 to 5 dB, in 5 dB steps). For the third VAD experiment, 4 of these 10 subsets (35, 25, 15 and 5 dB) were also included in the training material to train a new MLP, with the purpose of making the system more robust against noise. The files tested were the same as in the second experiment.

Finally, the results were compared using two standard VAD algorithms, and the same files as in experiments 2 and 3 were tested: the *Test* blocks of the *babble* noise and the *white* noise datasets of the *Noisy TIMIT* speech database.

Table 8.3 shows the main characteristics of the databases and the channels of each database used for this research.

Table 8.3: Main characteristics of the databases (and channels) used to develop the MNS-based VAD.

Database	*Basque Speecon-like*		*Spanish Speecon - ECESS*				*TIMIT*	*Noisy TIMIT*	
Channels used	Close-talk	Desktop mic	very close (C_0)	close (C_1)	med. (C_2)	far (C_3)	Headset-mounted and far-field mic	*babble* noise (50-5 dB)	*white* noise (50-5 dB)
Language	Basque		Spanish				English (USA)	English (USA)	
Environment	Office		Office, public place, entertainment, car				Studio	Studio + additive noise	
Speakers	230		60				630	630	
Files / speaker	316		17				10	600	
Total content (h)	109.95		1.41				5.37	322.2	
Speech content (%)	47.90		51.77				86.57	86.57	
Labelling	Phonetic Forced Alignment		Manually				Manually	From *TIMIT*	
Sample rate	16 kHz		16 kHz				20 kHz (down-sampled 16 kHz)	16 kHz	

8.6 MNS-based VAD experiments

The silence HMM was trained using the *Train* block of the *Basque Speecon-like* database. The audio signals were windowed into 25 ms length frames picking up a frame each 10 ms. Acoustic parameters include 13 MFCCs and 13 first and 13 second order derivatives, and they were modelled with 32 mixture GMMs. These parameters were normalised using the means and variances computed from the files belonging to the same session (all the utterances corresponding to the same speaker).

8.6.1 MNS-based VAD experiment using an MLP

In order to prepare the training data, six datasets were considered: the C_0, C_1, C_2 and C_3 from the *Spanish SpeeCon* database, and the *close-talk* and *desktop* from the *Basque Speecon-like* database. All the datasets consist of 1 020 files, which contain the same speech content if they belong to the same database. This means that the content of the 1 020 files of the C_0 dataset, for example, is the same as the content of the files of the C_3 dataset; and the content of the 1 020 files of the $close-talk$ dataset is the same as the content of the files of the *desktop* dataset. All the files have been processed to obtain the log-likelihood observation scores after normalising the acoustic features using the means and variances precomputed from each dataset. Thus, vectors of 6 scores are provided by each frame in each file. Altogether, 3 096 632 score vectors have been obtained, 49.08 % of which correspond to speech and 50.92 % to non-speech. This fact shows that the training data we have prepared are well balanced. Table 8.4 shows more information about the data.

The MLP used for this task contains 6 nodes in the input layer (one by each score) and 2 nodes in the output layer (one by each category). Half of the sum of both node amounts (4 nodes) has been chosen for the hidden layer.

Table 8.4: Datasets, number of files and number of frames considered to build the training data of the MLP.

	Basque Speecon-like db	*Spanish SpeeCon* db	
Datasets	*close*, *desktop*	C_0, C_1, C_2, C_3	TOTAL
Files	1 020 × 2	1 020 × 4	6 120
sil frames	299 972 × 2	244 211 × 4	1 576 788
speech frames	234 628 × 2	262 647 × 4	1 519 844
Total frames	534 600 × 2	506 858 × 4	3 096 632

The association of each training frame with a "speech" or "non-speech" category has been done, in the case of the *Spanish SpeeCon* database, using the labels provided by the database. These marks correspond to segments labelled as "sil" (non-speech), "u" (unvoiced speech) and "v" (voiced speech). Obviously, "u" and "v" segments have been used as category "speech". In the case of the *Basque Speecon-like* database, the marks have been obtained from the HMM training process, using the intermediate phone-alignment files.

For an initial testing of the MNS-based MLP, a separate database has been chosen: the *TIMIT* Acoustic-Phonetic Continuous Speech Corpus [196]. It is composed of a total of 6 300 utterances, 10 sentences spoken by each of 630 speakers from the 8 mayor dialect divisions of the United States of America. The 10 sentences represent roughly 30 seconds of speech material per speaker. In total, the corpus contains approximately 5 hours of speech. For practical reasons, some dialectal regions are less well-represented than the others. The same applies to female speakers, which are under-represented compared with male speakers: female, 30 %; male, 70 %.

Two-channel signals were recorded using a Sennheiser HMD 414 headset-mounted microphone and a Breul & Kjaer 1/2" far-field pressure microphone. The speech was directly digitised at a sample rate of 20 kHz with the anti-aliasing filter at 10 kHz and downsampled to 16 kHz. However, only the speech data recorded with the Sennheiser microphone was included on the CD-ROM release version, and that is the one we have used in this experiment.

The main advantage of using *TIMIT* is that it is perfectly labelled: segments marked as "h#" indicate initial/final silence, and the ones marked as "pau" indicate pauses. So the segments marked with these two labels have been considered as "non-speech"; the remaining segments have been labelled as "speech". All in all, considering 15 ms long frames at 10 ms frame rate, there is a total of 1 925 077 frames in the *TIMIT* database. Out of them, 86.57 % are "speech" frames, and only 13.43 % are "non-speech".

Two tests have been carried out in this experiment: on the one hand, all *TIMIT* files (6 300) have been processed in off-line mode, which signifies that each file's means and variances are computed beforehand. Then, acoustic features are normalised, and the threshold calculated in the previous section (Th = -150) is used to classify each frame as "speech" or "non-speech". On the other hand, the same *TIMIT* files have been processed in on-line mode; i.e. acoustic features are normalised on-line in different ways, obtaining a vector of 6 scores. These scores are classified on-line by the MLP. Off-line and on-line experiment results are shown in Table 8.5.

Table 8.5: TER, ER_0 and ER_1 of the off-line and on-line VAD experiment on *TIMIT* corpus.

	TER	ER_0	ER_1
Off-line experiment (Th = -150)	5.27	32.67	1.03
On-line experiment (MLP)	4.98	19.68	2.70

Results show that overall slightly better results have been obtained using the MNS-based MLP. Indeed, regarding TER, an improvement of 5.50 % has been achieved compared to the off-line experiment. The ER_0 has significantly improved as well. However, the ER_1 results have been worse, although they remain low.

It has to be taken into account that the threshold of the off-line experiment has been computed using exclusively the *Basque Speecon-like* database. Results of the experiments testing the C_0 subset of the *Spanish SpeeCon* database (Table 8.1, C_0 row) and the *TIMIT* database (Table 8.5) are therefore equivalent, since they test a dataset other than the one used to calculate the threshold. Results are similar except for ER_0, which shows a much higher value when testing the files from the *TIMIT* database. This fact is probably due to acoustic differences between databases. Nevertheless, the most important ratio, ER_1 remains low, and it is even lower in the *TIMIT* test.

With respect to the on-line experiment, the MLP has been trained using data from both *Basque Speecon-like* and *Spanish SpeeCon* databases, This may be the reason why better overall results have been obtained in this experiment, since data from more sources have been used. However, the MNS technique is an elaborated extension of the idea of calculating a threshold. This may likewise explain the improvement.

In conclusion, the experiment described in this section shows that the MNS technique proposed in this work is, at least, as efficient as the off-line technique. The main advantage of this technique is that it is able to classify a 15 ms length audio frame using a model, i.e. without the need for analysing the neighbouring frames or the frames of a segment (or file) to which it belongs. This is a very encouraging idea, because its implementation would not require any delay and would be really fast. Nevertheless, the results obtained in the experiment described in this section correspond to testing clean speech (the *TIMIT* database files), and consequently noisy speech must be tested in order to assess the robustness and real validity of this new technique. This is described in the next section.

8.6.2 MNS-based MLP experiments in noisy conditions

In order to assess the robustness of the technique introduced in the previous section, it must be evaluated testing noisy speech files. For that purpose, a database developed by the Florida Institute of Technology has been chosen: the *Noisy TIMIT* speech database [197]. This database contains approximately 322 hours of speech from the *TIMIT* database [196] modified with different additive noise levels. Therefore, its label files are the same as those belonging to the classic *TIMIT* database.

The additive noises are: babble, white, pink, blue, red, violet noise, with noise levels varying in 5 dB steps and ranging from 50 to 5 dB. The colour of noise refers to the power spectrum of a noise signal. Coloured noises are named in an analogy to the colours of light. For instance, white noise contains all audible frequencies just as white light contains all frequencies in the visible range. Instead, non-white coloured noises have more energy concentrated at the high or low end of the sound spectrum. All audio files are presented as single channel 16kHz 16-flac, but have been converted to 16-bit PCM (WAV).

The experiments consist of using the same MLP used in the previous section, but testing all the files of the *Noisy TIMIT* database. Here, 15 frames have also been considered as the minimum duration of both speech and silence segments. The TERs obtained for each kind of noise at different SNRs are shown in Figure 8.10, where the *TIMIT* baseline (Table 8.5) is shown as a dotted line.

Results show that each type of noise affects the VAD performance in a different way. At first glance, while red noise (a noise heavily weighted towards the lower end of the spectrum) hardly affects the overall results, white noise (flat frequency spectrum) and pink noise (decreases proportionally with increasing frequency) are the most damaging. Babble noise is the second less impacting noise (except for the case of 5 dB). However, due to the difference between the amount of analysed speech and non-speech frames (speech frames are 86.57 % of the total), the overall results tend to be similar to ER_1 results. Regarding ER_0 and ER_1, results can be seen in Figure 8.11, where the respective

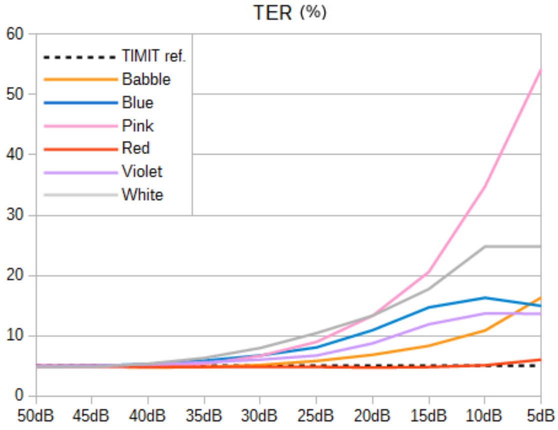


Figure 8.10: VAD TERs obtained testing the *Noisy TIMIT* database with the MNS technique, for different types of noise (colours) and SNR levels.

TIMIT baselines (Table 8.5) are shown as dotted lines.

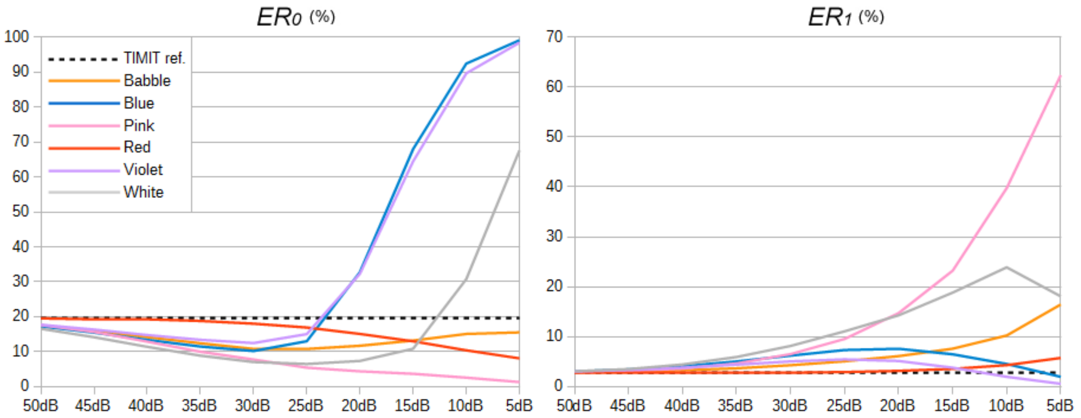


Figure 8.11: VAD ER_0s and ER_1s obtained testing the *Noisy TIMIT* database with the MNS technique, for different types of noise (colours) and SNR levels.

Analysing ER_0 and ER_1 results together, we can conclude that red noise is still the safer. With blue noise (noise that increases proportionally with increasing frequency) and violet noise (heavily weighted towards the higher end of the spectrum), the lower the SNR, the more non-speech frames labelled as speech. This means that there comes a point where all the frames are labelled as speech. The effect of this is the same as not using a VAD. Pink noise creates the opposite case. The lower the SNR, the more speech frames labelled as non-speech. This is the worst case for a VAD, since more and more speech frames are labelled as non-speech and are not therefore considered. With respect to white noise, characteristics of both cases are present. This means that classification is more unpredictable in that case.

Regarding babble noise, one could initially expect that the closer to 0 dB the SNR gets, the more non-speech frames classified as speech. However, results show a different behaviour: ER_0 shows a curve with a minimum of 10.75 % at 25 dB, with a value of 15.55 % at 5 dB (at 50 dB: 17.35 %); and ER_1 gets higher as SNR becomes lower. Misclassification of non-speech frames increases more significantly the closer SNR gets to 0 dB, which is unacceptable for a VAD system.

All in all, the MNS experiments in noisy conditions show a very irregular behaviour of the MLP classifier depending on the type of noise. However, it should be taken into account that the files used to train the MLP are not as noisy as the ones tested in this experiments. Thus, another experiment has been devised, in order to see whether the MNS-based MLP is able to model speech and non-speech frames at noisy conditions.

8.6.3 Including noisy signals in the MNS-based MLP

In this new experiment, noisy signals will be included at the training process of the MLP. Only babble and white noises have been taken into account, since they can be considered as the most natural ones for a system such as the one proposed in this thesis. So babble noise and white noise subsets of the *Noisy TIMIT* database have been added to the files from *Basque Speecon-like* and *Spanish SpeeCon* databases used in the MLP training process in the previous experiment. These noisy files belong to the *Train* block of each subset and, in each subset, 4 of the 10 different SNR signal groups have been used: 50, 35, 20 and 5 dB. This will give us insight into the need to use just a small set of noisy data.

Analysing the *Train* block, it has been found that there is a big difference between the amount of silence and speech frames: 190 052 *sil* frames and 1 220 017 *speech* frames. The amount of silence frames used from the *Spanish SpeeCon* database is 244 211 and from the *Basque Speecon-like* database is 299 972 (see Table 8.4). So, although a bit unbalanced, all the files of the *train* block have been used to keep the maximum number of silence frames. On the contrary, the number of speech frames have been randomly reduced to 244 003. Thus, this amount is comparable to the amounts used in Table 8.4. All in all, *Noisy TIMIT* contributes with 1 520 416 silence frames and 1 952 024 speech frames, being the total amount of frames 6 569 072 (3 097 204 *sil* frames, 3 471 868 *speech* frames). Table 8.6 shows all this information.

The score vectors now contain 14 elements: 2 scores corresponding to *close-talk* and *desktop* subsets of *Basque Speecon-like* database; 4 scores corresponding to C_0, C_1, C_2 and C_3 subsets of *Spanish SpeeCon* database; and 8 scores corresponding to the four different chosen SNR levels (50, 35, 20 and 5 dB) from the babble noise subset and white noise subsets of the *Noisy TIMIT* database. Having these score vectors of size 14, the MLP configuration selected for this experiment has been: 14 nodes in the input layer, 2 nodes in the output layer, and half of the sum of both node amounts (8 nodes) for the hidden layer.

Table 8.6: Datasets, number of files and number of frames considered to build the training data of the MLP with noisy signals.

	Basque Speecon-like	*Spanish SpeeCon*	*Noisy TIMIT*	
Datasets	*close*, *desktop*	C_0, C_1, C_2, C_3	*babble*, *noise*: 50, 35, 20, 5 dB	TOTAL
sil frames	299 972 × 2	244 211 × 4	190 052 × 8	3 097 204
speech frames	234 628 × 2	262 647 × 4	244 003 × 8	3 471 868
Total frames	534 600 × 2	506 858 × 4	434 055 × 8	6 569 072

The tested files are 10 080 files belonging to the *Test* block of babble noise and white noise subsets at all the available SNR levels (1 260 files in each SNR group). Results (TER) are shown in Figure 8.12 (solid lines), along with the TERs obtained in the previous experiment (dotted lines). At first sight, babble noise curve shows a small improvement, but white noise shows a significant improvement, especially as SNR gets smaller. At 5 dB, babble noise error rate is 12.15 %, which involves an improvement of 25.69 %, and white noise error rate is 8.44 %, involving an improvement of 65.87 %. It is noticeable that now both curves form a more similar shape.

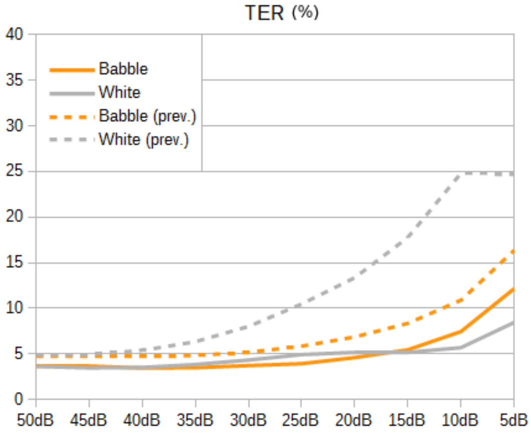


Figure 8.12: VAD TERs obtained testing the *Test* blocks of babble noise and white noise subsets of the *Noisy TIMIT* database with the MNS technique, for all the available SNR levels (dotted lines: result of the previous experiment).

Due to the difference between the amount of speech and non-speech frames in the *Noisy TIMIT* database, ER_0 and ER_1 must be analysed in order to make a more reliable and realistic interpretation. The ER_0 and ER_1 values are shown in Figure 8.13. Regarding ER_1, results are fairly similar to TER results, because of the unbalance explained before. So, the same explanation can be applied, further considering that results have improved at the cleanest conditions. At 5 dB, ER_1 of 11.09 % for babble noise and 7.52 % for white noise have been obtained. On the other hand, regarding ER_0, results have different impact on babble noise and white noise signals: the curve for the babble noise is similar to the previous one, but values are now a little worse except for the medium SNR values (around 25 dB). Regarding the white noise subset, the ER_0 curve also shows worse results for the cleanest part, but much better results at the noisiest conditions, which shot up from 15 dB on. The result is even better at 5 dB (14.51 %) than at 50 dB (18.26 %). It could well be said that the use of noisy data at the training process stabilises the results of testing noisy data.

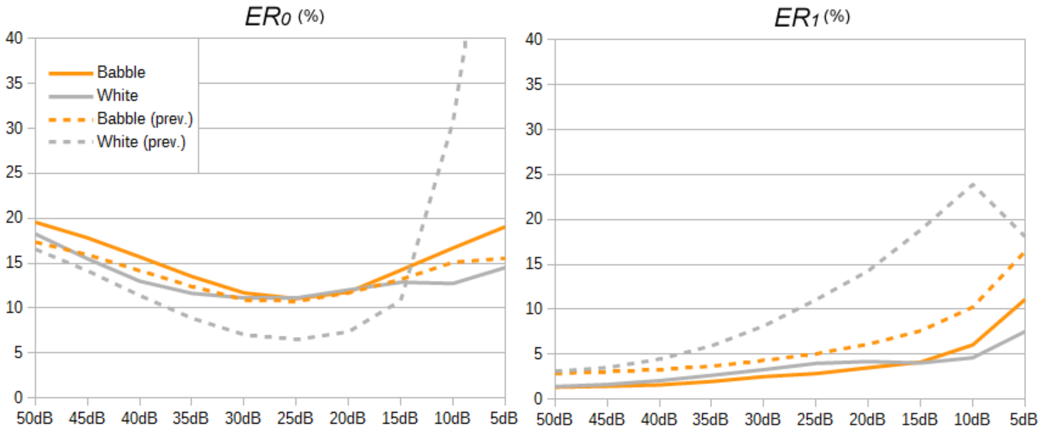


Figure 8.13: VAD ER_0s and ER_1s obtained testing the *Test* blocks of babble noise and white noise subsets of the *Noisy TIMIT* database with the MNS technique, for all the available SNR levels (dotted lines: results of the previous experiment).

It is worth noting that at the cleanest scenario (50 dB) an improvement of about 50 % is achieved, getting closer to the off-line result obtained in Table 8.5. This is a very competitive outcome (see Table 8.7).

Table 8.7: TER, ER_0 and ER_1 of the on-line VAD experiment on *TIMIT* corpus.

		TER	ER_0	ER_1
Off-line experiment (Th = -150)		5.27	32.67	1.03
On-line experiment		4.98	19.68	2.70
On-line exp. (+ noisy data)	babble noise 50 dB	3.73	19.57	1.32
	white noise 50 dB	3.63	18.26	1.40

In general, results show that, including some noisy data in the training process, similar ER_1 curves are obtained for babble and white noise. On the other hand, ER_0 results degrade a little for the cleanest signals but improve for the noisiest ones (improve substantially for white noise).

8.6.4 Generalisation to other types of noise

We have shown that the MLP trained with 4 subsets (the 35, 25, 15 and 5 dB) of each dataset *babble* and *white* is able to generalise results for the rest of SNR values. However, seeking to learn whether the MLP trained with noisy signals can also generalise for other types of noises, we tested the MLP trained with noisy signals (see section 8.6.3) with signals containing other types of noise. So now the test set comprises the files from the *Test* blocks of dataset *babble* and *white* along with the files belonging to the same *Test* blocks of the datasets *blue*, *pink*, *red* and *violet* (1260 files in each SNR subset; 12 600 in each dataset).

Figure 8.14 shows the TERs obtained at different SNRs for the various noise types. For SNRs equal to or greater than 35 dB there is no degradation when signals that have unseen noises are tested. For smaller SNRs, the deterioration is not very large: the maximum is for *violet* noise, which degrades by about 7 points at 15 dB with respect to both references. For *red* noise, the system actually behaves better than when the reference noises are tested.

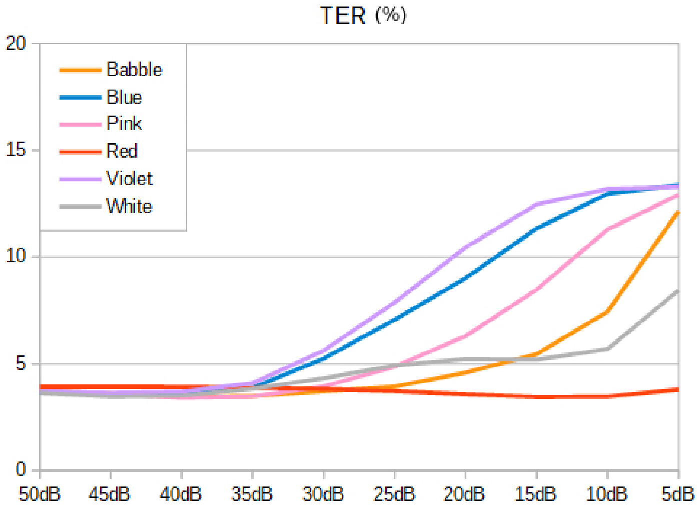


Figure 8.14: TERs obtained by testing all the SNR subsets of all types of noise of the Noisy TIMIT, using the MLP trained with signals containing *babble* noise and *white* noise.

8.7 Final experiments

Two on-line VAD algorithms standardised by ITU-T (International Telecommunication Union - Telecommunication Standardisation Sector) were tested to check the validity of the VAD technique proposed here. The algorithms belong to series G (*Transmission systems and media, digital systems and networks*), where *G.710 - G.729* are devoted to *Coding of voice and audio signals*. The first algorithm is *G.720.1* [198], which is actually a Generic Sound Activity Detector (GSAD) that can operate on 8 or 16 kHz audio input, with a VAD module. The second algorithm is *G.729* [199], an 8 kbit/s speech coder that manages 8 kHz input signals, which relies on a VAD module described in its Annex B (also known as *G.729b*). Both systems use a 10-ms frame length and frame shift, and no look-ahead is needed (no delay, just the frame duration). Further details are provided in Table 8.8 for both ITU systems[1] and our proposed VAD technique.

Note that the computation time is the average time per file needed by each system in a test where 10 080 files are processed, using the same computer and under the same conditions. It can vary from one computer to another, but it gives some idea of the ratios between them. Additionally, regarding the hangover scheme, the G.729b and our proposed VAD technique follow a similar state machine, and introduce a delay while it is decided whether there is a change or not. In the case of G.720.1 a conservative scheme is followed, where active indicators are emitted until a silence segment is detected.

Table 8.8: Comparison of some important parameters of the VAD in *G.720.1* (ITU-T), the *G.729b* algorithm (ITU-T) and our proposed VAD technique.

	G.720.1 VAD	*G.729b*	Prop. method
Bandwidth (kHz)	8, 16	8	16
Frame duration / shift (ms)	10 / 10	10 / 10	25 / 10
Computation time (ms per file)	26.8	34.87	30.7
Smoothing	No	Yes	Yes
Initialisation (No. frames)	200 inactive	32	0

To test the VADs, the same data were used as in section 8.6.2 and section 8.6.3. To test the G.729b coder VAD, the files had to be down-sampled to 8 kHz. Figure 8.15 shows the silence misclassification error rates (ER_0) obtained using the two ITU algorithms (solid lines) and our proposed method (dotted lines), both for babble noise and white noise signals. The figure shows that our proposed system let, at most, 20 % of silence frames pass as speech. The minimums of both ITU systems are over 30 %, and they show a significant increase as SNR gets lower, especially for babble noise signals. This means that more and more silence frames are classified as speech as signals get noisier.

On the other hand, Figure 8.16 shows the speech misclassification error rates (ER_1). In the case of G.720.1, babble noise signal results and the results of our proposed method

[1] The software for both systems can be downloaded from the ITU website: `http://www.itu.int/rec/T-REC-G.720.1-201001-I` and `http://www.itu.int/rec/T-REC-G.729-201206-I`, respectively.

are actually quite similar. Regarding white noise, results are similar as well for low-noise signals, but our proposed method gets better results on the noisiest data. In the case of G.729b, our proposed method obtains better results for SNR higher than 25 dB, and then G.729b algorithm shows a better performance. At 10 and 5 dB, G.729b gets better results for babble noise, and the MNS-based VAD for white noise.

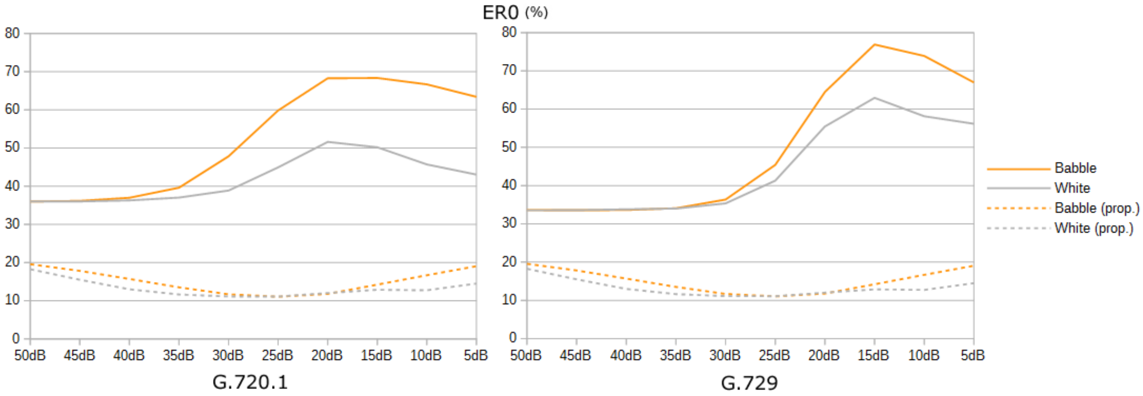


Figure 8.15: VAD ER_0s obtained using the ITU-T G.720.1 (left) and G.729b (right) standard VADs on babble noise and white noise signals from *Noisy TIMIT*, along with the results of the proposed system (dotted lines).

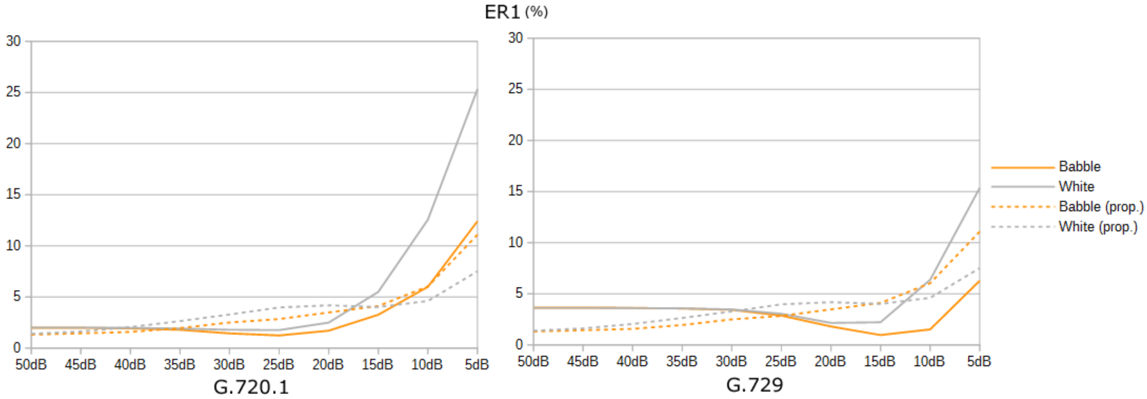


Figure 8.16: VAD ER_1s obtained using the ITU-T G.720.1 (left) and G.729 (right) standard VADs on babble noise and white noise signals from *Noisy TIMIT*, along with the results of the proposed system (dotted lines).

In general, ER_1 results obtained by the ITU algorithms and the MNS-based VAD are comparable for high SNR signals. By contrast, for low SNR signals the results show different behaviour for *babble* noise and *white* noise. For *babble* noise, G.720.1

gets similar results, and G.729b gets better results. For *white* noise, better results are obtained by the MNS-based system. Nevertheless, it is worth noting that ER_0 values are very high for the two ITU algorithms, which means that both systems tend to classify non-speech frames as speech when testing noisy signals.

In conclusion, our proposed VAD technique gets better TER at all noise levels. Due to the imbalance between the amount of speech and non-speech frames, the TER curves are similar in shape to those obtained for ER_1 but are shifted proportionally by ER_0. One of the advantages of the ITU systems is that they can adapt to different noise conditions on-line; however, they need an initialisation time to adjust the main parameters. In comparison, our MNS-based system is able to generalise for noise types that are not included in the training process and it requires no initialisation time, since the results do not depend on any previous frame.

8.8 Conclusions

In this chapter, a new VAD technique has been described: the use of observation scores generated by the central-state GMM of the silence HMM. These scores have proved to efficiently discriminate between speech and non-speech audio frames. Simply by calculating a threshold, very competitive results are obtained. However, in order for the VAD system to operate properly, audio features must be normalised beforehand, using the means and variances corresponding to the audio segment (or file) that is being analysed. This makes the system off-line, because the VAD analysis cannot start until the whole data has been processed.

To overcome this issue, an extension of that threshold-based technique has been devised: the multi-normalisation scoring or MNS. This idea is based on normalising the acoustic features using means and variances computed from different datasets, so that we can model the behaviour of the observation scores they produce. Training a MLP as a classifier, we have got similar results to those obtained with the threshold-based technique. For clean speech results are highly competitive. However, for noisy speech, results show an irregular behaviour depending on the type of noise. Including some noisy signals among the training data can alleviate the problem in some extent. Indeed, regarding ER_1, improvements are achieved at all SNR levels, but the error rates obtained at the noisiest level (5 dB) remain rather high. Better results could be probably obtained if more noisy data were used when training the MLP.

A validation experiment has been also performed, comparing the results of two ITU-T VAD algorithms with our MNS-based VAD. The performance of our proposed VAD technique when it is trained with noisy signals (*babble* noise and *white* noise) from different SNRs is better overall than the performance of the ITU-T standard systems G.720.1 and G.729b, since the classification error is considerably lower for non-speech and is comparable for speech segments. This makes our technique useful for both systems that require low speech error rates and systems that require low non-speech error rates. Furthermore, our VAD seems to generalise the results properly for intermediate SNRs and the unseen noise types tested, which makes the system robust to different noise levels and types.

One of the greatest advantages of the MNS-based technique is that it performs on-line, making decisions frame by frame, with no need to analyse the neighbouring frames or the frames of a segment (or file) to which it belongs. In addition, the use of observation likelihoods as the basis of a VAD is also interesting due to its great simplicity. In a system where HMMs are used (as in an ASR system), the proposed VAD requires very little extra processing. The main disadvantage could be how the VAD behaves with unseen noises: it seems to be able to generalise results, but the error rate increases somewhat at some SNRs. Further research is needed to determine how the system could perform a proper generalisation.

Future research directions could be the analysis of the observation likelihoods obtained from a *speech* GMM, the analysis of the generalisation of results when processing audio signals containing unseen noises, the test of different classifiers in addition to MLPs —as Recurrent Neural Networks (RNN)— and the test of the system in real-world conditions.

CHAPTER 9

On-line CMVN

9.1 Introduction

The performance of ASR and ASR-based systems degrades if the acoustic conditions of the training and test signals do not match. The techniques used to date can be broadly classified into two categories: model adaptation and feature normalisation. Model adaptation techniques transform the trained models aiming to match the test utterance conditions, whereas feature normalisation techniques modify the test features in order to match the statistics of the training features. Feature normalisation techniques can be further categorised as parametric and non-parametric approaches. In this thesis we focus on parametric feature normalisation techniques; specifically, on Cepstral Mean and Variance Normalisation (CMVN).

Cepstral Mean Normalisation (CMN) was initially proposed for feature normalisation [200]. CMN matches the first order moment of every utterance by removing their respective time average and transforming each utterance to zero mean. It is supposed to compensate the channel effects in the form of convolutional noise. At the end of the '90s CMVN became very popular [201] [202] [203]. It matches both mean and variance by transforming every utterance to zero mean and unit variance. Since CMVN was devised, different improvements have been developed based on CMVN, such as the use of CMVN in model domain [204], the use of CMVN combined with SNR features [185], the use of posterior estimates of mean and variance instead of the maximum likelihood estimates (Bayesian approach) [205], the use of Maximum-Likelihood Mean and Variance Normalisation (ML-MVN) to estimate the early mean and variance vectors in an utterance (CMVN is applied later, from a threshold amount of frames on) [206] etc. Each new technique has its own advantages and drawbacks, but nowadays it is common practice to perform CMVN in the field of robust speech recognition [207].

In the next section (section 9.2) we discuss the fundamentals of CMVN. The impact of noise and channel distortions on clean speech is examined. In section 9.3 three different implementations of on-line CMVN are described, which use causal recursive updating. Experimental results of each implementation are also shown. In section 9.4, a new on-line normalisation approach is presented (directly related to the MNS technique explained in section 8.4): the MNS-based CMVN. Finally, some conclusions are discussed in section 9.5.

9.2 Fundamentals of CMVN

In order to understand the meaning of applying CMVN, suppose that we have an input signal $x[n]$ and suppose that the channel impulse response is given by $h[n]$. The final signal is the linear convolution of both (see equation (9.1)).

$$y[n] = x[n] * h[n] \tag{9.1}$$

By taking the Fourier Transform, due to its convolution-multiplication equivalence property, we get equation (9.2).

$$Y[f] = X[f] \cdot H[f] \tag{9.2}$$

In order to calculate the cepstra, we take the logarithm of the spectrum (see equation (9.3)).

$$\log Y[f] = \log\left(X[f] \cdot H[f]\right) = \log X[f] + \log H[f] \tag{9.3}$$

We now return to the time domain (or, better, to the q quefrency domain) through the inverse FT, obtaining thus the cepstral coefficients (see equation (9.4)).

$$Y[q] = X[q] + H[q] \tag{9.4}$$

It can be seen that, in cepstral domain, any convolutional distortions are represented by addition. Let us assume that the distortions suffered by the speech signal are stationary (which is a rather realistic assumption, as vocal tract and channel response do not change in very short segments) and the speech signal can be viewed as a piecewise stationary signal (or as a short-time stationary signal). Then, for every i-th (stationary) frame, equation (9.5) is fulfilled.

$$Y_i[q] = X_i[q] + H[q] \tag{9.5}$$

By taking the average over all frames, we get equation (9.6).

$$\frac{1}{N}\sum_i Y_i[q] = \frac{1}{N}\sum_i X_i[q] + H[q] \tag{9.6}$$

Defining D_i as the difference between the cepstral value at i-th frame (Y_i) and the mean, it can be written as in equation (9.7).

$$D_i[q] = Y_i[q] - \frac{1}{N}\sum_j Y_j[q] = X_i[q] + H[q] - \left[\frac{1}{N}\sum_j X_j[q] + H[q]\right] = \tag{9.7}$$

$$= X_i[q] - \frac{1}{N}\textstyle\sum_j X_j[q]$$

In looking at equation (9.7), it may be concluded that removing mean values from the cepstra, we remove the channel distortions.

For noisy scenarios, considering additive noise, equation (9.1) becomes equation (9.8), and equation (9.2) becomes equation (9.9).

$$y[n] = x[n] * h[n] + w[n] \tag{9.8}$$

$$Y[f] = X[f] \cdot H[f] + W[f] \tag{9.9}$$

And taking the logarithm, we obtain equation (9.10).

$$\log Y[f] = \log\left[X[f]\left(H[f] + \frac{W[f]}{X[f]}\right)\right] = \log X[f] + \log\left(H[f] + \frac{W[f]}{X[f]}\right) \tag{9.10}$$

Now, the term $\frac{W[f]}{X[f]}$ appears, which can be negligible in low-noise conditions, but also very important in poor SNR conditions.

Summing up, Cepstrum Mean Normalisation (CMN) can compensate for convolutional distortions. Furthermore, it has been demonstrated that the cepstral mean characterises not only the channel transfer function, but also the average frequency response of the vocal tracts of different speakers. In consequence, by removing the long-term speaker average, CMN can act as a sort of speaker normalisation [135]. On the other hand, variance normalisation (CVN) is not associated with addressing a particular type of distortion. However, it provides robustness against acoustic channels, speaker variability, and additive noise [201]. CMVN takes advantage of both techniques.

The most common approach tries to estimate per feature mean and variance vectors over an utterance [183] or over a windowed-out part of an utterance (usually for real-time systems) [203]. The feature vectors are then shifted and scaled by the estimated means and variances aiming at zero mean and unity variances for the normalised features. For N cepstral vectors $y = \{y_1, y_2,..., y_N\}$, their mean μ and variance σ^2 vectors are calculated as defined in equation (9.11) and equation (9.12), respectively.

$$\mu_N(i) = \frac{1}{N}\sum_{n=1}^{N} y_n(i) \tag{9.11}$$

$$\sigma_N^2(i) = \frac{1}{N}\sum_{n=1}^{N} (y_n(i) - \mu_N(i))^2 \tag{9.12}$$

where i is the i-th component of the vector. The cepstral features are then normalised using the calculated mean and variance vectors, as given in equation (9.13).

$$\hat{y}_n(i) = \frac{y_n(i) - \mu_N(i)}{\sigma_N(i)} \tag{9.13}$$

9.3 A study on CMVN for on-line implementation

The best recognition results with CMVN are usually obtained when the values of MFCC means and variances are previously estimated by averaging along the entire current utterance and are kept constant during the normalisation process [208]. In the CAPT task, audio samples can be sent packaged in a *wav* file, so CMVN can be easily applied processing initially the whole file and normalising then the cepstral features of that file. However, in the Word-by-Word Sentence Verification (WWSV) task, normalisation must be carried out as soon as audio samples come in, and hence means and variances can not be computed using the whole audio segment. An initial estimation of both means and variances is needed, as well as a subsequent update or adaptation.

Different approaches are used in the literature to normalise cepstral parameters on-line:

- **Using past data**: In this approach the means and variances of the features are not estimated using the current utterance, but other data sources, as signals from the current session or set of sessions, or even the previous signal. Then the signals are normalised with these constant values. This method has proved to have very good results, as shown in [209], but it has the disadvantage that past information must be available.
- **Segmental approach**: It was firstly proposed in [203], where MFCC vectors are normalised using CMVN over a sliding finite length normalisation window, in which the analysed frame is located at the centre. In isolated-word recognition experiments, the best performance is achieved for windows of about 1 s, which means that a delay of 0.5 s is added, too long to be applied for a WWSV system.
- **Recursive approach**: The mean and variance vectors are initialised using the first D frames of the current utterance and then they are recursively updated as new frames arrive [210]. It is therefore a non-causal system where the frame used to update the estimations is D frames ahead of the frame being normalised (hence, it is called *look-ahead parameter* D), as shown in equation (9.14) and equation (9.15). Obviously, the actual delay produced by the normalisation technique is D. Note that $x_n(i)$ denotes the ith feature vector component at frame n, and β is a forgetting factor.

  $$\mu_n(i) = \beta \cdot \mu_{n-1}(i) + (1-\beta) \cdot x_{n+D}(i) \tag{9.14}$$

  $$\sigma_n^2(i) = \beta \cdot \sigma_{n-1}^2(i) + (1-\beta) \cdot (x_{n+D}(i) - \mu_n(i))^2 \tag{9.15}$$

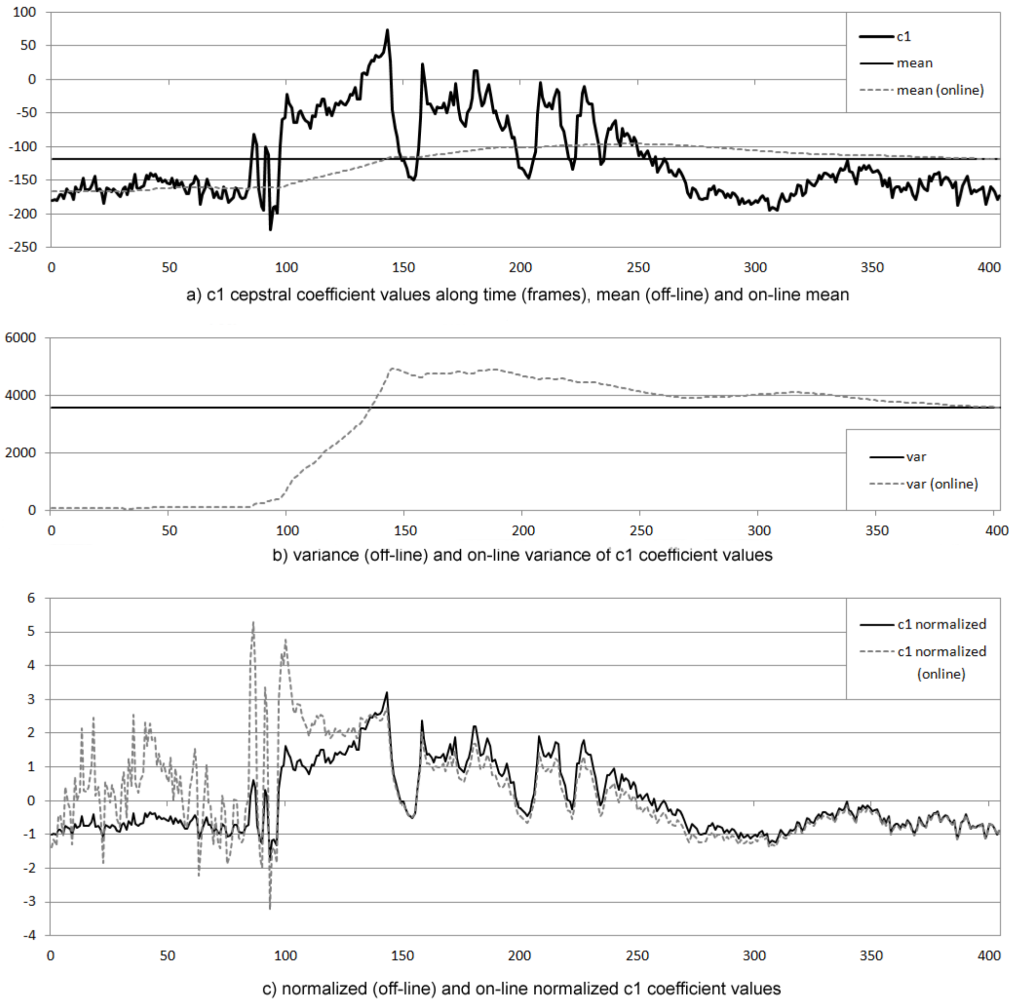
  For the same delay, better results than in segmental approach are obtained with the recursive approach.

  Recursive approach can be improved by better estimating the initial values of means and variances. For example, using past data to estimate the initial values and recursively updating them, results very close to those obtained off-line can be achieved. If past data is not available, data from the first non-speech-to-speech transition can be used, which gets better results than using segments from the beginning of the utterance [209]. However, this approach requires a VAD.

9.3.1 Different implementations of on-line CMVN

In this section, we will see the impact of three on-line normalisation techniques. In the first, the initial means and variances are estimated using the first D frames and then updated using the incoming frames. In the second, the initial values are computed beforehand, from the train database. In the third, a hybrid approach is implemented, where the initial means are estimated using the first D frames and then updated frame by frame, and the variances are estimated from the train database and kept constant.

a) Initial look-ahead and recursive updating



a) c1 cepstral coefficient values along time (frames), mean (off-line) and on-line mean

b) variance (off-line) and on-line variance of c1 coefficient values

c) normalized (off-line) and on-line normalized c1 coefficient values

Figure 9.1: Initial look-ahead and recursive updating: a) $c1$ coefficient (thick black), off-line mean (black) and on-line mean (dashed) with 250 ms look-ahead; b) $c1$ off-line variance (black) and on-line variance (dashed); c) Resulting normalised $c1$ values for off-line (black) and on-line calculation (dashed).

In this approach, the initial values of means and variances are calculated using the first D frames. These D frames are normalised using the estimated mean and variance values, and from then on, means and variances are updated using the current frame, as shown in equation (9.16) and equation (9.17).

$$\mu_n(i) = \frac{(n-1)\mu_{n-1}(i) + x_n(i)}{n} \tag{9.16}$$

$$\sigma_n^2(i) = \frac{(n-1)\sigma_{n-1}^2(i) + (x_n(i) - \mu_{n-1}(i))(x_n(i) - \mu_n(i))}{n} \tag{9.17}$$

Figure 9.1 shows an example of the effects of using this technique: at the top picture, the values of the second parameter ($c1$) along time (10 ms separated frames) are shown, along with the off-line mean value (constant) and the recursively updated on-line mean value. At the centre, the off-line variance curve and the on-line curve. At the bottom, the resulting normalised $c1$ curves in the off-line case and the on-line case.

The figure shows that a good initial estimate of means and variances is key to ensure an appropriate coefficient normalisation. Calculating the initial value of variance using only non-speech frames, a very low value is obtained, which causes a big impact when normalising the coefficient values. Mean does not seem to have as great an impact.

Phonetic experiment results are presented in section 9.3.2.

b) Using past data

The use of previous data to estimate the initial values of means and variances seems to be very effective if the current audio signal's characteristics are similar to those previous ones. A typical mode of obtaining these values is using the current session or a set of sessions. Although using the current session (recordings belonging to the same user) better results are obtained, especially for medium distances, it has the drawback of the initialisation. Using a set of sessions achieves slightly worse results, but more data can be available [209].

In this section, we will estimate the global values of means and variances using the *Basque Speecon-like* database. Two scenarios have been considered:

- **Constant mode**: precalculated values are kept constant. The results are very intuitive: the curve of normalised coefficients will be shifted and amplified proportionally with the difference between the means and variances of the training signals and those of the current audio signal.
- **Updating mode**: precalculated values are recursively updated using the frame being analysed. It is necessary to check whether the normalised coefficients are stable or not.

Figure 9.2 graphically illustrates an example of the *updating mode*. The effect of the precalculated initial values of both the mean and variance lasts a very short time:

the mean curve joins immediately the raw on-line curve in a few frames; the variance curve needs some frames more, but it immediately drops close to zero, which means that coefficient values are being highly amplified. This fact brings us to the previous implementation, and hence similar results are expected.

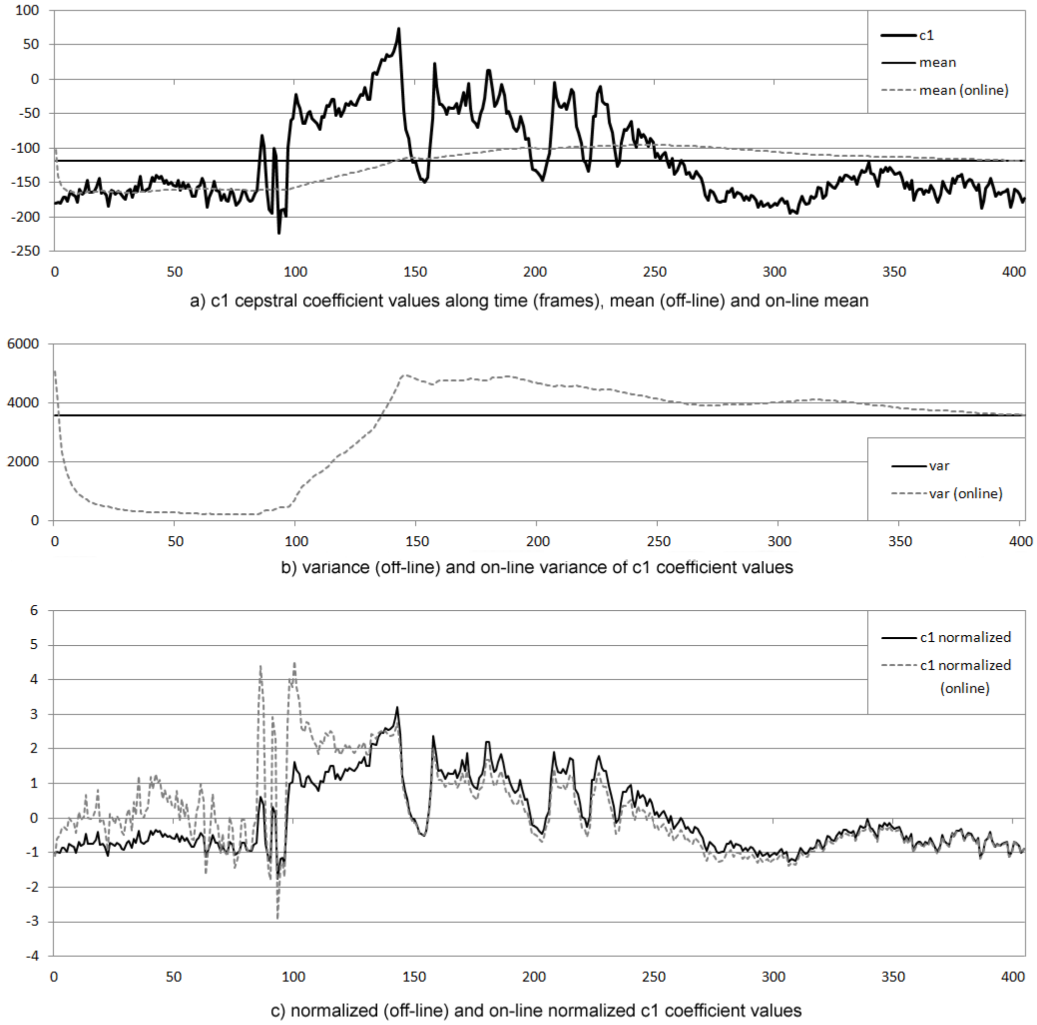


Figure 9.2: Past data as initial estimation and recursive updating: a) $c1$ coefficient (thick black), off-line mean (black) and on-line mean (dashed); b) $c1$ off-line variance (black) and on-line variance (dashed); c) Resulting normalised $c1$ values for off-line (black) and on-line calculation (dashed).

Phonetic experiment results of both the *constant* mode and the *updating* mode using past data are presented in section 9.3.2.

c) The hybrid approach

In the two previous implementations, it has become evident that a bad estimation of the initial values of means and variances has a negative impact on the final results. Furthermore, a proper estimation of the variance is even more crucial, since normalised coefficients are computed dividing by the standard deviation (square root of the variance).

In this new approach, characteristics of the previous two implementations have been used, hence its name: hybrid approach. On the one hand, it will take advantage of the look-ahead approach to estimate the initial mean values. After D look-ahead frames, the means will be recursively updated. On the other hand, past variance data will be used in *constant* mode, without updating. Figure 9.3 illustrates this graphically.

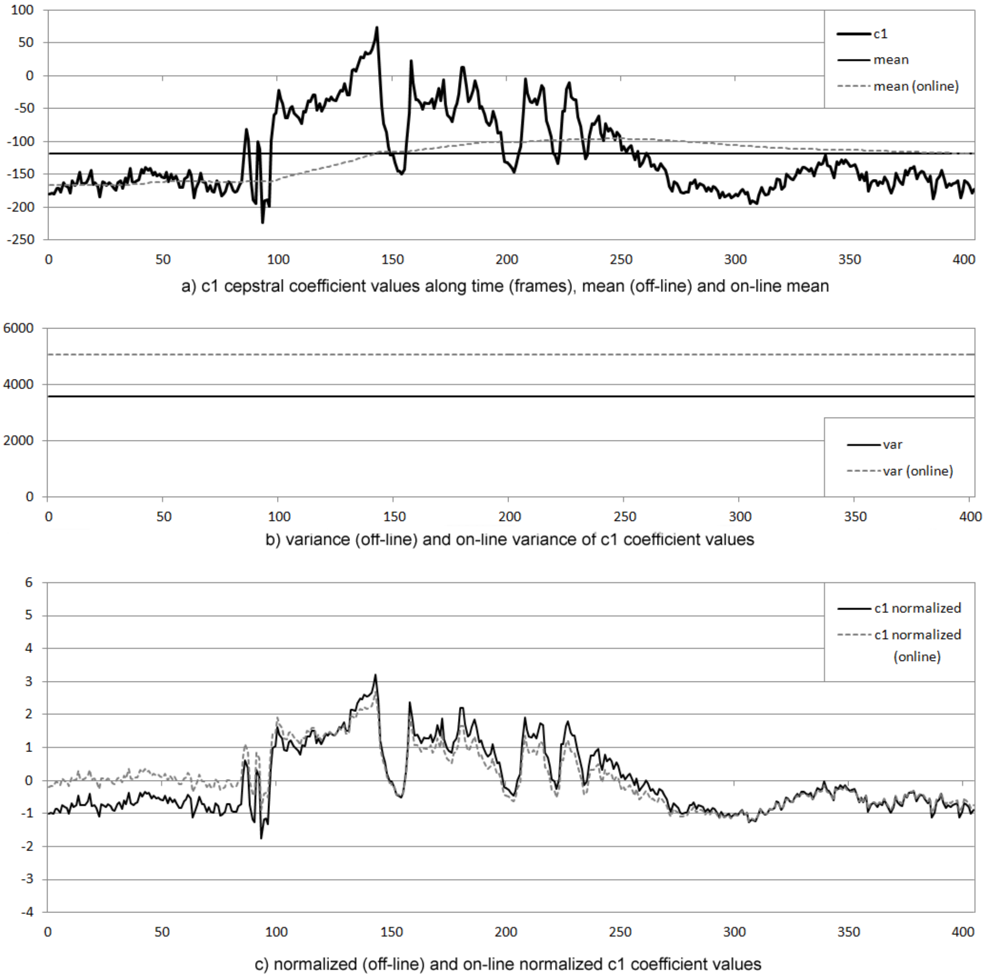


Figure 9.3: Hybrid approach: a) $c1$ coefficient (thick black), off-line mean (black) and on-line mean (dashed); b) $c1$ off-line variance (black) and on-line variance (dashed); c) Resulting normalised $c1$ values for off-line (black) and on-line calculation (dashed).

In the hybrid approach, the problem of initial low variance values is removed, which helps avoid initial distortions. At the same time, mean values are initially estimated from the first D frames and subsequently updated. This ensures quite a uniform cepstral normalisation, except at the beginning, where normalised coefficient values can be shifted due to the difference between means. However, this issue does not seem to have a decisive impact on silence recognition; furthermore, if the first D frames include speech frames, better estimation of means would be performed. The main drawback of this approach is that the variance of the audio signal being analysed must not differ significantly from that previously computed.

Phonetic experiment results of the hybrid approach are presented in the next section.

9.3.2 Experimental results

In order to evaluate the distortion added to the system by this on-line initial mean and variance estimation method, the phonetic test carried out in section 5.3 has been repeated, but using the different on-line implementations described in the previous section instead of the off-line one. The HMMs used for the test have been the ones giving the best result in the off-line test. Audio files belonging to the *close-talk* subcorpus (recorded via head-mounted microphone) as well as files corresponding to the *desktop* subcorpus (recorded by means of a microphone located at a distance of 1 m) have been tested, with the intention of gaining a broader view. Table 9.1 shows the results obtained in these tests for each on-line implementation mode and for each different channel, in terms of Phone Error Rate (PER, %) and Accuracy (%). Off-line results have been also displayed in the table, to make it easier for the reader to interpret the results.

Table 9.1: Results of the three different on-line implementations: PER and Accuracies with close-distance and 1 m distance signals, compared with off-line values.

	PER (%)				
	Off-line	On-line			
	Current data	Look-ahead + update	Past data	Past data + update	Hybrid
Close-talk	12.45	18.39	14.20	18.29	15.30
Desktop	21.87	29.92	29.98	29.62	33.38

	Accuracy (%)				
	Off-line	On-line			
	Current data	Look-ahead + update	Past data	Past data + update	Hybrid
Close-talk	74.24	38.91	59.00	42.09	54.05
Desktop	60.12	22.13	37.04	26.72	43.79

The results of Table 9.1 show that, as expected, PERs and Accuracies in all the on-line cases are worse than the off-line ones. Using *past data* keeping these precalculated means and variances constant for the entire signal, slightly worse PERs are obtained when testing *close-talk* files, but much worse when testing *desktop* files, because of the channel mismatch. Accuracy, which takes into account insertions, substitutions and deletions, is also considerably worse in both cases, especially for *desktop* channel files, where a big deterioration is achieved. Nevertheless, even having so poor results, this approach provides the best results for *close-talk*.

The updating approaches (the *look-ahead + update* and the *past data + update*) show very similar results. If we focus on Accuracy, very low results are obtained for both approaches (the *past data + update* approach shows slightly better results though). This worsening mainly happens initially, while just non-speech frames are being considered in the computation of the variances. The impact of this is the insertion of many phones instead of silence, since those initial frames are boosted due to the low variance values of silence.

The hybrid approach gets the best results among the approaches that use current data, except for PER in *desktop* files. It has got the second best PER when testing *close-talk* channel files but the worst for *desktop* files. In terms of accuracy, the second best result has been obtained as well for *close-talk* files, and the best result for *desktop* files, but still very poor.

All in all, using past data is a simple solution to the problem, but results degrade considerably when testing signals recorded through a different channel, since no adaptation is produced to the current signal's characteristics. On the other hand, updating means and variances frame by frame not very good results are obtained, mainly because non-speech and speech frames must be included in a proper initial estimation of means and variances. An hybrid approach has also been tested, where, on the one hand, *look-ahead* is used to estimate the initial mean values, and after D look-ahead frames, these means are recursively updated; on the other hand, past variance data is used in constant mode, without updating. This approach seemed to take advantage of both approaches, but results are not good.

9.4 MNS-based CMVN

9.4.1 Introduction

In section 8.4, we have introduced a novel VAD technique, which is based on the MNS method proposed in this thesis. This method consists in generating multiple observation likelihood scores, by normalising the incoming MFCCs using means and variances computed from different speech datasets recorded under different conditions. Thus, a set (or vector) of observation likelihoods is obtained, which follows different patterns depending on the nature ("speech" or "non-speech") and the SNR of the signal frame being analysed.

For the VAD, a Multi-Layer Perceptron (MLP) has been used to classify the score vectors obtained with MNS into two classes: *speech* and *non-speech*. However, the objective now is to find out which dataset a frame belongs to (which would introduce no delay). If the MLP could give us this information, we could use the global means and variances computed from the corresponding dataset to normalise the MFCCs of the incoming signal. Thus, assuming that we have N databases available —each of which has its own global MFCC μ_n mean and σ_n^2 variance vector—, the new MLP will provide, as a result, the probabilities of belonging to each dataset. The main idea of the MNS-based CMVN is to estimate the means and variances of each of the incoming frames ($\hat{\mu}(i)$ and $\hat{\sigma}^2(i)$, respectively, for each i^{th} feature vector component) as the weighted sum of the means and variances computed from each dataset, using the corresponding probabilities as w_n weights. This is shown in equation (9.18) and equation (9.19).

$$\hat{\mu}(i) = \sum_{n=1}^{N} w_n \cdot \mu_n(i) \tag{9.18}$$

$$\hat{\sigma}^2(i) = \sum_{n=1}^{N} w_n \cdot \sigma_n^2(i) \tag{9.19}$$

The new MLP will contain the same number of inputs and outputs. The input to the MLP will be a score vector of N elements, and the output will be the probability of belonging to each of the N datasets that we are using.

9.4.2 MNS-based CMVN experiments

To test the validity of the proposed MNS-based CMVN method, we have to assess whether the estimated MFCC means and variances are appropriate. Firstly, we have done an initial phonetic recognition experiment estimating the means and variances as described in the introduction. Different datasets have been considered for MNS to train the MLP, with the aim of creating a general MLP model.

After the initial experiment, an analysis of the results has been done, in order to see the differences between the estimated values of means and variances using only silence frames and the estimated values using all the frames. This has led us to carry out another experiment, using not only the observation likelihoods obtained from the central state of the silence HMM trained using the *close* channel from the *Basque Speecon-like* database, but also the ones obtained from a new GMM trained with all the speech data. Thus, a new MLP has been trained and the initial experiment has been repeated.

Finally, conclusions are drawn.

a) Initial phonetic recognition experiment

For the initial experiment, a MLP has been trained with the observation likelihood vectors obtained using the MNS method on 12 datasets: the *close* and *desktop* channels from *Basque Speecon-like* database [101], the C_1 and C_3 channels from the subset of the *Spanish Speecon* database used in an ECESS evaluation campaign of voice activity and voicing detection [180] and 8 datasets corresponding to the *Noisy TIMIT* speech database [197], specifically 4 datasets from the *babble* noise channel and other 4 from the *white* noise channel, corresponding to 50, 35, 20 and 5 dB values of SNR. All the datasets have been taken from the *train* part of each database, letting a *test* part for the experiments.

The HMMs used for the phonetic recognition are the ones created following the process "R + *M25*", using the single-entry dictionary and 32 Gaussians. Note that the best results applying off-line CMVN with these HMMs was a PER of 12.45 % (see section 5.3).

The results of the experiment are shown in Table 9.2. Comparing with the values presented in Table 9.1, the PER obtained for *Close-talk* in this experiment is the third best one, but the PER obtained for *Desktop* is the lowest. Regarding the accuracies, the best values are those obtained in this experiment.

Table 9.2: Results of the implementation of the MNS-based CMVN (on-line): PERs and Accuracies with close-distance and medium-distance signals, compared with off-line values.

	PER (%)	
	Off-line	On-line
	Current data	MNS-based
Close-talk	12.45	15.51
Desktop	21.87	29.29

	Accuracy (%)	
	Off-line	On-line
	Current data	MNS-based
Close-talk	74.24	65.50
Desktop	60.12	45.45

b) Data analysis

To check the similarity of the estimated means and variances with the real ones, we have carried out another experiment: a new MLP has been trained using the observation likelihood vectors obtained from 8 datasets: the 50, 35, 20 and 5 dB channels of the

Train blocks of the *babble* and *white* noise subsets of *Noisy TIMIT* database. With this MLP, all the channels (the *Test* blocks) have been tested: 50 to 5 dB, in 5 dB steps. Thus, we have obtained the probabilities of each input vector to belong to each *Train* subset (8 subsets), and then we have applied equation (9.18) and equation (9.19) to estimate the MFCC means and variances.

Two tests have been done: on the one hand, MFCC means and variances have been estimated considering only silence frames. On the other hand, MFCC means and variances have been estimated considering all the frames. The reason why this distinction has been made is that the observation likelihoods are calculated by means of a silence GMM, so it is reasonable to assume that silence frames will be better classified by the MLP.

Figure 9.4 shows the 0^{th} MFCC (top diagrams) and the 1^{st} MFCC (bottom diagrams) computed for the various noise levels of *babble* noise. The green curves in the figures correspond to the average values of real means of the tested files. The orange curves represent the average values of the estimated means taking into account all the frames of the tested subsets; and the red curves depict the average values of the estimated means computed using only silence frames of the tested subsets. Clearly, the curves obtained using only silence frames are more similar to the real curves. It may be due to the fact that a greater variability of the MNS score vectors is shown at speech frames than at silence frames (see Figure 8.8 in section 8.4), and as a consequence it is more difficult to model the vectors corresponding to speech frames.

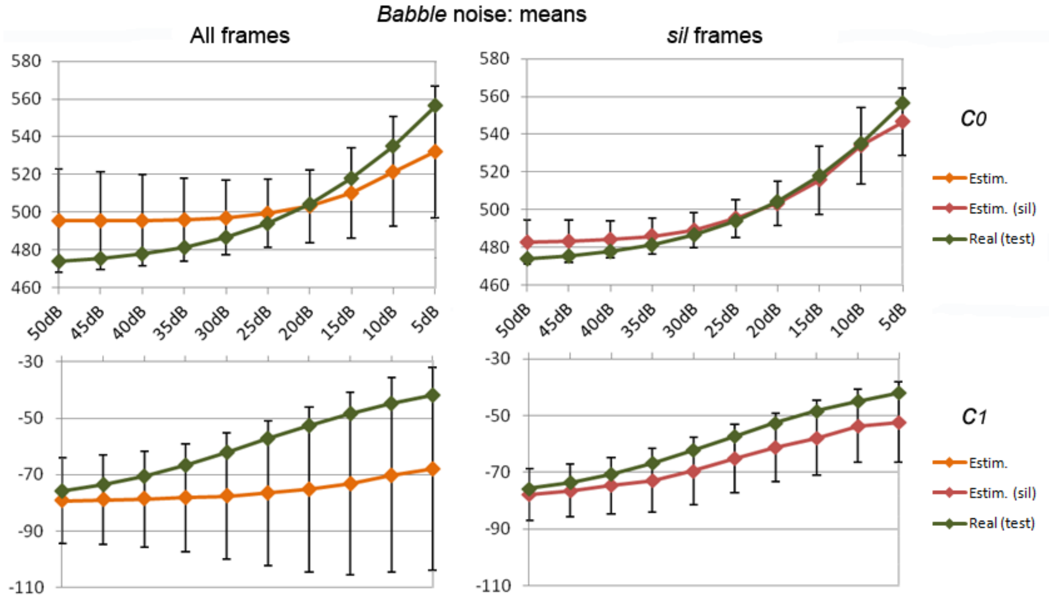


Figure 9.4: Real values vs. estimated values (left: using all frames; right: using sil frames) of 0^{th} (top) and 1^{st} (bottom) MFCC means and standard deviation, at different noise levels for *babble* dataset.

Real and estimated variances of the 0^{th} and 1^{st} MFCC are also shown in Figure 9.5 for *babble* noise, reaching the same conclusion: comparing the curves of the real values (green) with the curves of the estimated variances computed using all the frames (orange) and only silence frames (red), the ones obtained using only silence frames are more similar to the curves of real values.

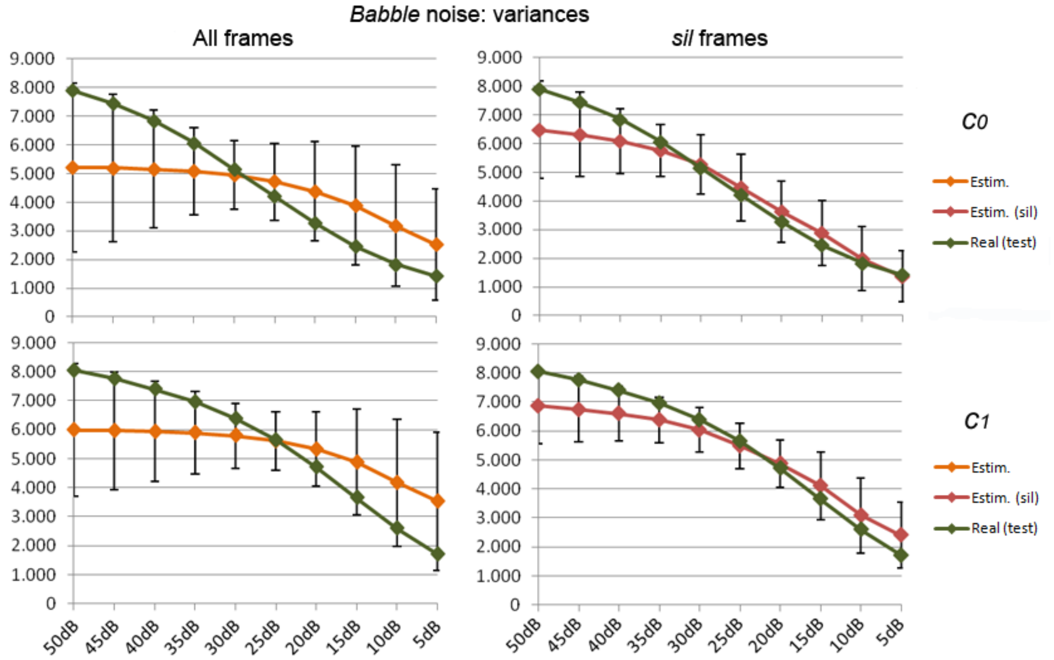


Figure 9.5: Real values vs. estimated values (left: using all frames; right: using sil frames) of 0^{th} (top) and 1^{st} (bottom) MFCC variances and standard deviation, at different noise levels for *babble* dataset.

The same conclusion has been obtained for *white* noise channel signals.

c) Phonetic recognition experiment including a *speech* GMM

To see whether the speech frames can be better modelled, observation likelihoods generated by a *speech* GMM would be also necessary. Our observation likelihood vectors are created using a *silence* GMM, trained using the *close-talk* channel from the *Basque Speecon-like* database. Thus, a *speech* GMM has been trained using all the speech frames of the *Train* block of the *close-talk* channel of the *Basque Speecon-like* database.

A new MLP has been trained using the same datasets as in the initial experiment. Thus, the vectors to train the new MLP contain 12 observation likelihoods provided by the *silence* GMM and other 12 provided by the *speech* GMM. The new MLP contains therefore 24 inputs, 12 outputs (one for each dataset considered) and a hidden layer with 18 neurons. With this new MLP, the initial experiment has been repeated. Results are shown in Table 9.3 along with the results of the initial experiment.

Table 9.3: Results of the implementation of the MNS-based CMVN including a *speech* GMM (on-line): PERs and Accuracies with close-distance and medium-distance signals, compared with off-line values.

	PER (%)		
	Off-line	On-line	
	Current data	MNS-based (sil GMM)	MNS-based (sil + speech GMM)
Close-talk	12.45	15.51	13.18
Desktop	21.87	29.29	25.28

	Accuracy (%)		
	Off-line	On-line	
	Current data	MNS-based (sil GMM)	MNS-based (sil + speech GMM)
Close-talk	74.24	65.50	71.36
Desktop	60.12	45.45	50.22

In this last experiment, es expected, better PER and accuracy results have been obtained using a *speech* GMM together with the *silence* GMM to generate observation likelihoods and feed the MLP using MNS. If we compare these results and the results of Table 9.1, we see that the best *on-line* results are obtained by the MNS-based method using both the *silence* GMM and the *speech* GMM. PER results are better than accuracy values, and results are also better for the *close-talk* channel than for the *desktop* channel. Summing up, the MNS-based normalisation can be an alternative to all *on-line* normalisation methods presented in this chapter. This new method works fine with clean signals, the error introduced is not very large. However, the error for noisier signals is bigger.

9.5 Conclusions

In this chapter, on-line CMVN has been discussed. The best recognition results with CMVN are obtained when the values of MFCC means and variances are previously estimated by averaging along the entire current utterance (the off-line approach). On-line CMVN has the disadvantage that some normalisation has to be applied before the entire signal has arrived, and means and variances have to be estimated following another approach. Currently it is still a challenging issue.

The classic implementations can be classified in three approaches: *using past data*, where means and variances are estimated using previously picked up signals; *the segmental approach*, where MFCC vectors are normalised applying CMVN over a sliding window; and the *recursive approach*, where mean and variance vectors are initialised using the first D frames of the current utterance and then recursively updated as new frames arrive.

Considering these three approaches, four implementations have been tested:

- **Using past data**: The means and variances computed from the *close-talk* channel of the *Basque Speecon-like* database are used to normalise the whole signals.
- **Using past data and recursive updating**: The means and variances computed from the *close-talk* channel of the *Basque Speecon-like* database are used as the initial values of means and variances, and then they are updated with the values obtained from the incoming frames.
- **Initial look-ahead and recursive updating**: D frames are used to compute the initial values of means and variances, and then they are updated frame by frame.
- **The hybrid approach**: D frames are used to compute the initial values of means and then they are updated frame by frame; the variances are computed from the *close-talk* channel of the *Basque Speecon-like* database, which are constant.

These implementations depend heavily on the distribution of speech throughout the signal. For example, much better results will be obtained if the look-ahead includes both non-speech and speech frames. Besides, if updating mode is used, long silence segments or long speech segments unbalance the means and variances and this has a great impact on the results. This means that these implementations are unstable.

Our proposed MNS-based on-line CMVN overcomes such disadvantages. The MNS-based method has no delay, because it does not depend on previous or future frames. The results obtained using the MNS-based CMVN are much better. Besides, it does not depend on the distribution of speech throughout the signal; so it behaves in the same way for any speech and non-speech distribution in the signal. The biggest drawback is that the method loses accuracy for noisy signals. The MNS-based on-line CMVN need more research for robust normalisation, but still it seems to be very useful.

CHAPTER 10

Phone scoring: from GOPs to DNNs

10.1 Introduction

In Chapter 6 preliminary experiments about phone scoring have been described. Goodness of Pronunciation (GOP) scores have been chosen as verification scores, and the way to obtain the GOP scores of incorrectly pronounced phonemes has been to simulate errors artificially inserting changes in the pronunciation dictionary. This relies on the idea that a phoneme is more incorrectly pronounced as its realisation is closer to the realisations of another phoneme in a particular language.

The thresholds are obtained calculating the Equal Error Rate (EER) point between the GOP distributions calculated for the correctly pronounced phonemes and the incorrectly pronounced ones (with the technique of simulating errors). Thus, an EER point was computed for each phoneme group. Although the laboratory tests produced very good results, an experiment in a realistic environment showed that the performance of the system got worse. Actually, there were specific phones that hardly exceeded the threshold. Besides, initial and final phones do not behave in the same way as the other phones, so specific GOP distributions would be needed to set their GOP thresholds. The position of the phone in the word seems to be an important characteristic to be taken into account.

The method of using a set of GOP thresholds to decide whether a phoneme has been correctly or incorrectly pronounced is somewhat limited. Current works take into account phone durations, posterior probabilities, GOPs etc. [66]. Besides, the information about adjacent phones can also be valuable, since a mispronounced phoneme would have an impact not only on the current phone, but also on the adjacent phones, not only in their GOPs, but also in their durations and probabilities.

In linguistics, a *phoneme* can be defined as "a set of speech sounds that form a basic contrastive unit in a language. [...] The members (or allophones) do not contrast with one another in the language in question" [211]. So phonemes create minimal pairs in a particular language, and consequently we can conclude that the pronunciation correctness of a phoneme is directly related to the rest of the phonemes, i.e. it could be understood in terms of the distance —in a N-dimensional space— between the realisations of a phoneme and the realisations of the rest of the phonemes in the context of a particular language.

10.2 The concept of incorrectly pronounced phoneme

A phoneme in a particular language can be considered as incorrectly pronounced, if its realisation is somehow closer to the typical realisations of another phoneme rather than to its typical realisations. Thus, as can be seen in Figure 10.1, if we assume that the realisations of a phoneme are points in a N-dimensional space, a new realisation of a phoneme p_1 could be considered as "correctly pronounced" if it is located closer to the typical realisations (or centroid) of phoneme p_1 compared with the realisations of the rest of the phonemes. If the realisation of p_1 is located closer to the realisations (or centroid) of another phoneme, it could be considered as "incorrectly pronounced". As has been explained before, these distances or intervals are variable, and depend on the distance to the realisations of the rest of the phonemes.

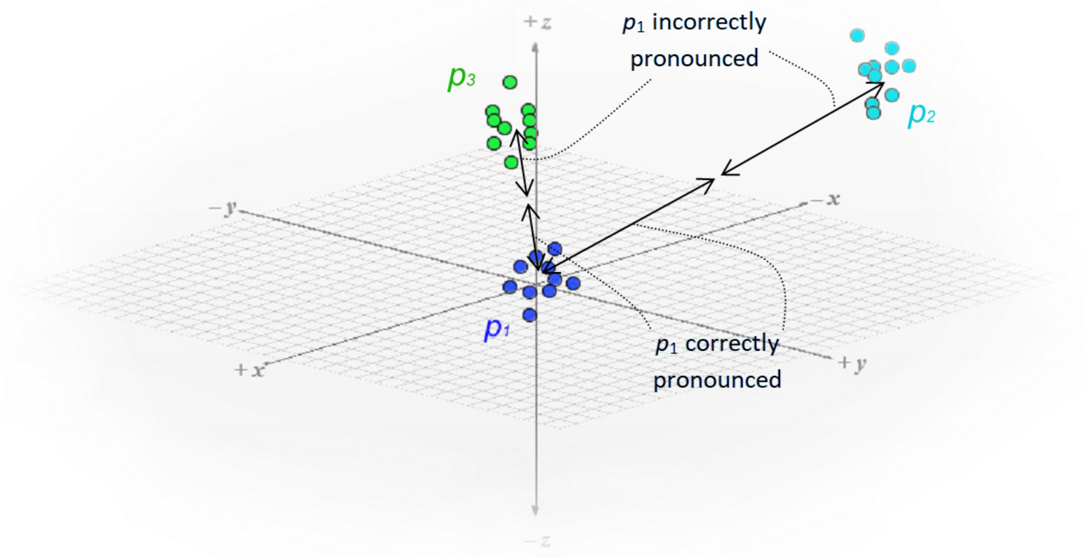


Figure 10.1: Realisations of phonemes p_1 (dark blue), p_2 (light blue) and p_3 (green) in a 3-dimensional space.

From this point of view, how correctly or incorrectly a phoneme is pronounced depends on the phoneme inventory of a particular language. For example, if a language contains 12 oral vowels (see SAMPA French[1]) and another one contains 5 (Basque) then the space for "correctly pronounced" phonemes will be bigger around a Basque vowel than around a French vowel. In short, the pronunciation correctness depends on the rest of the phonemes that belong to the language inventory and the distance between them.

[1] https://www.phon.ucl.ac.uk/home/sampa/french.htm

10.3 Training data

10.3.1 The training dataset

The training data have been chosen so that speech data include the lowest number of irregularities and mismatches. As in the first experiments of this work (see section 6.2), only the eastern Basque native speakers have been used from the *R+M25* subset (i.e. subset R —*Read* part— plus subset $M25$ —25 sessions *Manually* corrected) of the *Train* part of the *Basque Speecon-like* database to obtain the training data (76 speakers out of 155), since in this area Basque fricative sibilants, on the one hand, and the three affricate sibilants, on the other, are differently uttered.

In total, there are 761 503 phones in that audio subset. Half of them (50.03 %) are vowels; on the contrary, the phoneme group with the lowest representation is the group of palatals (1.24 %). A breakdown of the number of phones in the chosen data by group is shown in Table 10.1. It is remarkable the small amount of data available for palatals (L, jj and gj) and affricates (tz, ts and tS). It will have an impact on the results, which is more serious in the case of the affricates, since it usually is one of the most problematic group for Basque students.

Table 10.1: Number of phones (%) in the training data grouped by phonetic groups.

	No. of phones (%)
Vowels	50.03
Fricatives	6.89
Affricates	2.09
Voiced plosives	9.14
Unvoiced plosives	12.72
Nasals	7.62
Palatals	1.24
Liquids	10.27

10.3.2 Obtaining incorrectly pronounced phones

Taking into account the concept described in the previous section about what an incorrectly pronounced phoneme is, we can consider that, within a language, a completely incorrectly pronounced phoneme consists in pronouncing another phoneme of the same language. The method of simulating errors (described in section 6.2) fits this idea, since it obtains information (in that case, GOPs) replacing phones in the pronunciation dictionary of the system. Thus, the recogniser, in forced alignment mode, obtains the GOPs computed from the HMM of the replaced phone, considering that it is being pronounced as the original phone (a simulation of being incorrectly pronounced). Notice that phones were replaced only with those phones belonging to the same acoustic group, so that results are more conservative.

In this experiment, we have repeated the error simulation process, but using a bigger amount of audio data: each phone has been replaced by the other phones belonging to its same group and processed each time, so that a bigger amount of information is obtained for the simulated errors, specifically the information of 2 591 755 phones. In order to balance the amount of correctly pronounced and incorrectly pronounced data, many simulated errors have been discarded. If the number of phones obtained by simulating errors was bigger than the number of correctly pronounced phones, some incorrectly pronounced phones were discarded until it equals the number of correctly pronounced phones. By contrast, if the number of incorrectly pronounced phones was less than the number of correctly pronounced phones, both numbers were left as they were. The final amounts are shown in Table 10.2 broken down by the position of the phones.

Table 10.2: Total number of phones (C: correctly pronounced; X: incorrectly pronounced, simulated error) available in the training data regarding their position in the utterance and grouped by phonetic groups.

	Left phones		Middle phones		Right phones		Sole	
	C	X	C	X	C	X	C	X
Vowels	26 812	26 812	308 982	308 982	44 742	44 742	499	
Fricatives	6 995	6 995	42 916	42 916	1 147	1 147	30	
Affricates	367	157	14 911	9 534	684	684	0	
Voiced plosives	14 835	10 495	54 609	54 609	126	114	0	
Unvoiced plosives	6 549	6 549	84 618	79 736	5 618	3 725	0	
Nasals	3 732	2 071	38 657	38 657	3 090	3 090	0	
Palatals	529	529	8 000	6 492	305	305	0	
Liquids	1 609	1 609	73 747	73 747	930	930	0	

10.3.3 Data analysis

In section 6.2 GOP scores were extracted from correctly and incorrectly pronounced phones, and GMMs were trained for each phoneme group. However, durations and likelihoods have been also retrieved in this experiment. In addition, GOPs, durations and likelihoods of the surrounding phones have also been taken into consideration, in order to see whether they provide useful information.

In this section, we show some histograms of GOPs, durations and log-likelihoods, for the phone being studied and both its previous and posterior phones. Thus we could get a general idea about what each element's contribution could be.

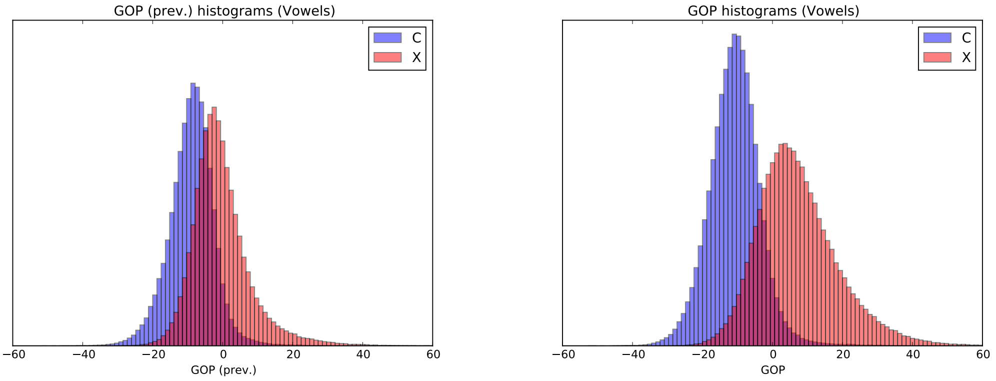


Figure 10.2: GOP histograms (right) and previous phone's GOP histograms (left) obtained from correctly (C) and incorrectly (X) pronounced vowels.

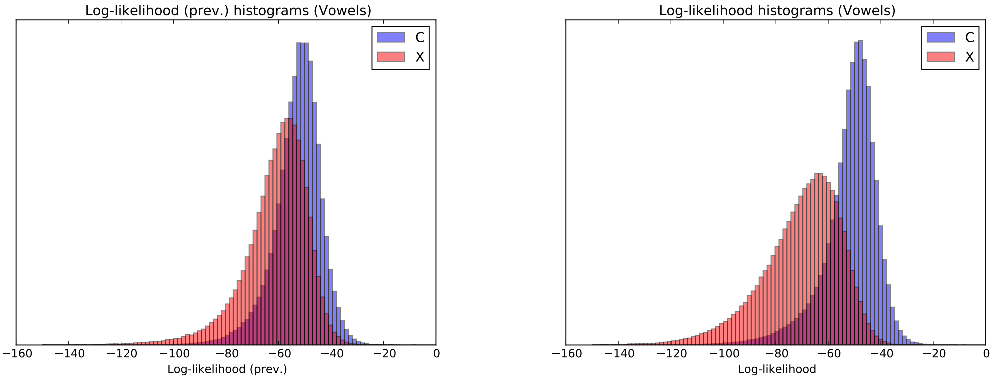


Figure 10.3: Log-likelihood histograms (right) and previous phone's log-likelihood histograms (left) obtained from correctly (C) and incorrectly (X) pronounced vowels.

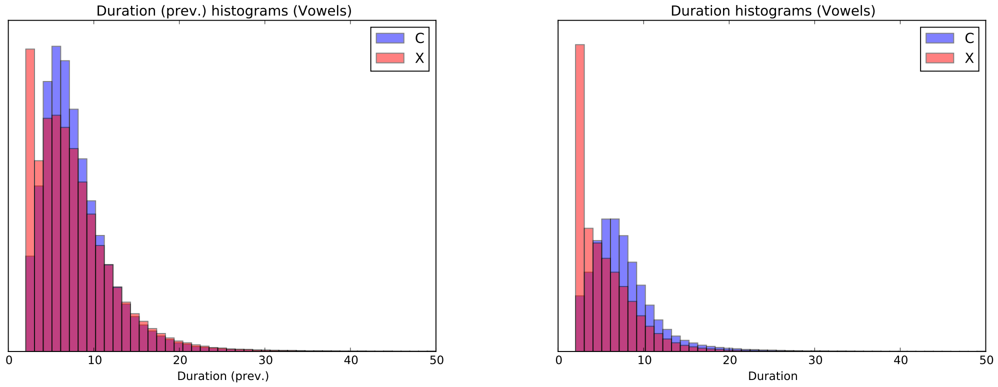


Figure 10.4: Duration histograms (right) and previous phone's durations histograms (left) obtained from correctly (C) and incorrectly (X) pronounced vowels.

Let us see the most and less favourable cases. The most favourable case is formed by vowels, specifically the middle ones. Figure 10.2 shows the GOP histograms of vowels for the current one (right) and its previous phone (left). While the current phone's GOP distributions seem quite discriminative, the previous phone's distributions are very overlapped. Concerning the log-likelihood (see Figure 10.3), the same conclusion as for GOPs can be drawn. And regarding the durations (see Figure 10.4), both distributions are very overlapped, but even more for the previous phone. Summing up, it seems that the previous phone does not provide discriminative information, and neither do durations.

On the other hand, the most unfavourable cases are formed by palatals, affricates and, to a lesser extent, fricatives. For palatals, all the histograms are almost completely overlapped (see Figure 10.5 for GOPs and Figure 10.6 for log-likelihoods). This is due to the confusion between these phones (see explanation in section 5.2.1). Besides, this

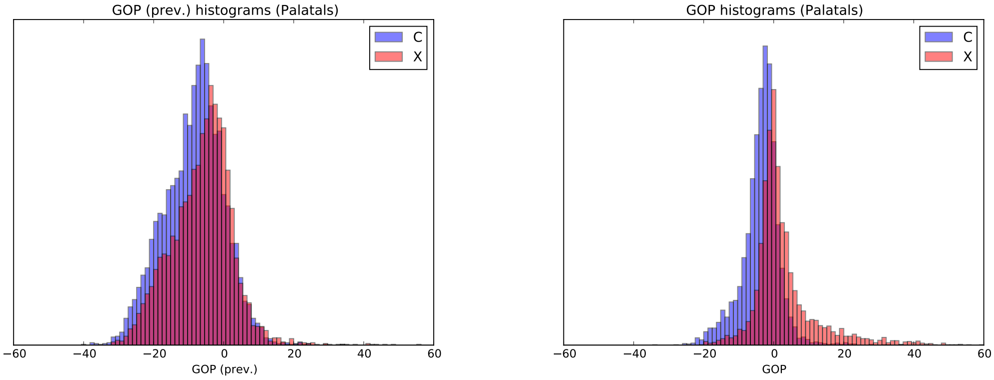


Figure 10.5: GOP histograms (right) and previous phone's GOP histograms (left) obtained from correctly (C) and incorrectly (X) pronounced palatals.

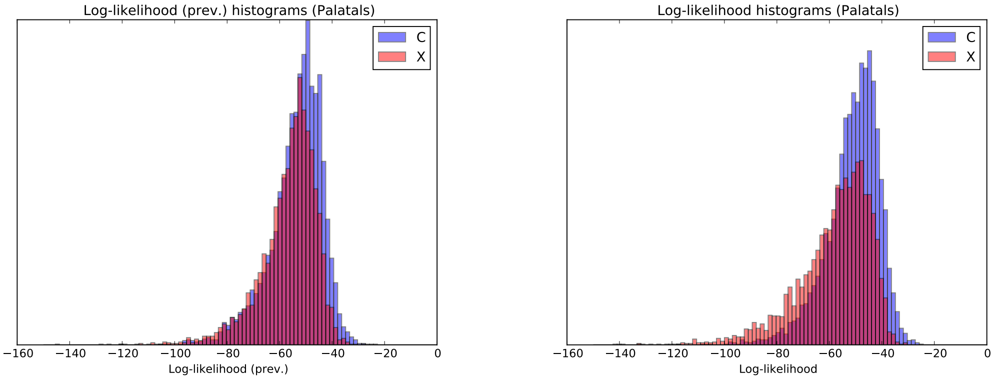


Figure 10.6: Log-likelihood histograms (right) and previous phone's log-likelihood histograms (left) obtained from correctly (C) and incorrectly (X) pronounced palatals.

phoneme group faces the difficulty of having the lowest number of instances in the *Basque Speecon-like* database.

In the case of affricates (Figure 10.7, right), the histograms are also quite overlapped (note that using not only the eastern native speakers of the database but also the non-native speakers the distributions overlapped completely). For fricatives the histograms are a little bit more separated (Figure 10.7, left).

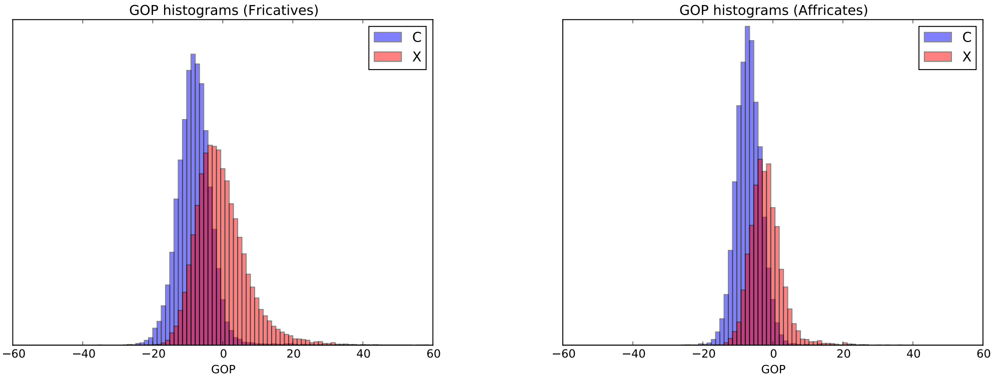


Figure 10.7: GOP histograms obtained from correctly (C) and incorrectly (X) pronounced fricatives (left) and affricates (right).

Regarding the rest of the groups (voiced plosives, unvoiced plosives, nasals and liquids), GOP histograms seem quite useful to discriminate. Log-likelihood histograms look more overlapped. As an example, the GOP and log-likelihood histograms of voiced plosives are shown in Figure 10.8.

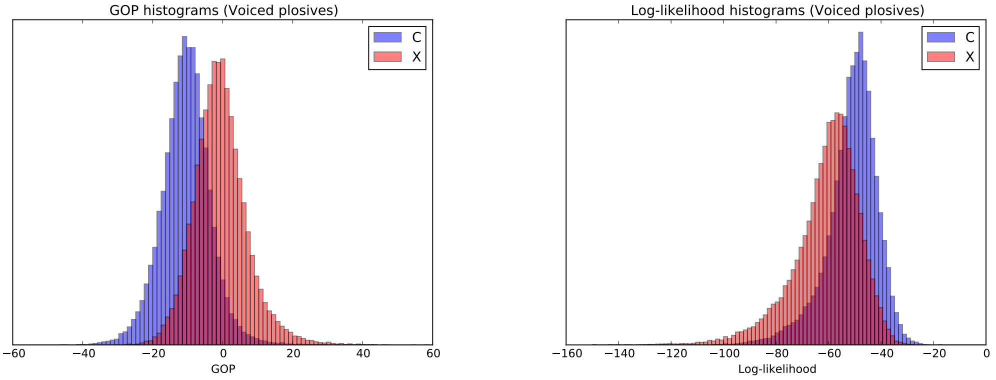


Figure 10.8: GOP histograms (left) and log-likelihood histograms (right) obtained from correctly (C) and incorrectly (X) pronounced voiced plosives.

All the histograms are collated in appendix B, broken down by phoneme group and position.

10.4 The decision making: Neural Networks

As explained in the conclusions of [36], where a summary of existing research on automated pronunciation error detection is done, one of the largest remaining challenges is to integrate current components into one that ideally is L1-independent, or at least easily configurable for a different L1, without requiring a manually annotated non-native database. Following this idea, we will use Neural Networks (NN) to see whether it is feasible to create a general composite model, and we will see what impact different parameters (GOPs, durations and likelihoods) have on the NN model.

10.4.1 Training parameter sets

Different Multi-Layer Perceptrons (MLP) have been trained in order to see if they can be used to make the decision on whether a phoneme is correctly pronounced or not. Firstly, a unique MLP has been trained for all the phones, indicating in the input layer the identity of each phone. Then, different MLPs have been also trained for each phone group.

Besides, different parameter sets have been used to train the MLPs:

a) GOPs, durations and log-likelihoods + previous and posterior phones' GOPs, durations and log-likelihoods.

b) GOPs, durations and log-likelihoods.

c) GOPs and log-likelihoods.

d) GOPs.

e) GOPs of previous, current and posterior phones.

MLPs have been created using TensorFlow™, an open source software library for numerical computation using data flow graphs, with a flexible architecture to deploy computation to one or more CPUs or GPUs. TensorFlow was originally developed by researchers and engineers working on the *Google Brain* team within *Google*'s *Machine Intelligence* research organisation for the purposes of conducting machine learning and deep neural networks research. Nowadays it is mainly used to train and test NN of different configurations [212].

10.4.2 Test data

To test the different models introduced in the previous section, we have used the audio signals that eventual users have recorded in the last years using our on-line CAPT demonstrator[1]. 30 people left their recordings in our server, and no user information is provided. Apparently, all the users' native language is Basque or Spanish. There are 24 males (56 files) and 6 females (22 files), which form an average of 2,6 files per speaker. All kinds of background noises, SNRs and recording devices can be found in them.

[1] https://aholab.ehu.es/users/igor/CAPT

The files have been manually labelled. In total, there are 1269 phones, from which 29 phones are incorrectly pronounced. Real data has the disadvantage that usually there are far fewer incorrectly pronounced phones than correctly pronounced ones, which unbalances the analysis of the results. Note that a binary decision has been made here, so that a phone can only be correctly or incorrectly pronounced. The amount of incorrectly pronounced phones is shown in Table 10.3 by groups. Almost 80 % are sibilants, both fricatives (z, s and S) and affricates (tz, ts and tS), something to be expected, since users' native language is Basque or Spanish. On the other hand, no plosive has been incorrectly pronounced.

Table 10.3: Number of incorrectly and correctly pronounced phones in the test data.

	Incorrect	Correct
Vowels	2	633
Fricatives (sibilants)	13	125
Affricates	10	83
Voiced plosives	0	75
Unvoiced plosives	0	103
Nasals	2	113
Palatals	1	14
Liquids	1	94

10.4.3 Results

The metric used to get results is the *Scoring Accuracy* (SA). The SA is computed as in equation (10.1),

$$SA\,(\%) = \left(\frac{CA + CR}{CA + CR + FA + FR}\right) \cdot 100 \tag{10.1}$$

where: CA: Correctly Accepted units; CR: Correctly Rejected units; FA: Falsely Accepted units; FR: Falsely Rejected units.

Table 10.4 shows the SA results obtained by each MLP in each test. The MLPs of this table are trained using a hidden layer of 64 nodes. Table 10.5 shows the same results for a hidden layer of 6 nodes in the MLPs.

Table 10.4: Scoring Accuracies (SA) obtained in each test by each MLP trained with a different parameter set and 64 nodes in the hidden layer.

	Tests (Scoring Acc.)				
MLPs	a	b	c	d	e
Single model	57.76	64.46	62.81	**65.09**	58.79
Vowels	63.62	69.61	69.76	**74.02**	65.83
Fricatives	50.00	**62.32**	55.05	60.14	49.28
Affricates	35.48	**45.16**	40.86	40.86	37.63
Voiced plosives	50.67	52.00	54.67	**61.33**	49.33
Unvoiced plosives	42.72	46.60	42.72	**51.46**	43.69
Nasals	65.22	74.78	**80.87**	77.39	**80.87**
Palatals	60.00	60.00	**66.67**	**66.67**	53.33
Liquids	67.37	**70.53**	65.26	69.47	68.42

Table 10.5: Scoring Accuracies (SA) obtained in each test by each MLP trained with a different parameter set and 6 nodes in the hidden layer.

	Tests (Scoring Acc.)				
MLPs	a	b	c	d	e
Single model	60.91	65.41	59.47	**67.13**	59.73
Vowels	65.83	70.71	69.67	**82.36**	65.35
Fricatives	52.90	68.12	60.87	**69.57**	50.00
Affricates	38.71	43.01	39.78	**52.69**	36.56
Voiced plosives	49.33	54.67	57.33	**70.67**	49.33
Unvoiced plosives	44.66	40.78	42.72	**52.43**	42.72
Nasals	65.22	79.13	79.13	78.26	**80.00**
Palatals	46.67	66.67	66.67	**73.33**	66.67
Liquids	64.21	67.37	65.26	**69.47**	61.05

The best results are obtained using 6 nodes in the hidden layer and using the MLP trained only with current GOPs. In the test of 6 nodes, almost all the best results are obtained in test d (current GOPs), which means that, in general, the other parameters introduce noise. In the case of nasals, previous and posterior GOPs improve the result, but not in the rest of the cases.

For 64 nodes in the hidden layer, results are more shared: 4 of the best results are obtained in test d, but 3 are obtained in test b, 2 in c, and 1 in e. Here, results can be interpreted as if each group had its own best parameter group and the 64 nodes of the hidden layer were able to model these differences. However, the results are lower than those obtained for 6 nodes, except for nasals, palatals and liquids. Another test has been carried out in order to see if more nodes in the hidden layer were beneficial for the models, training models with 128 hidden nodes, and the worst results were obtained.

Note that SA values are very close to CA values because of the unbalance between the number of incorrectly and correctly pronounced phones. So, in order to have a more detailed picture of the behaviour of the results, we have provided a table for the best test (test d with 6 nodes) showing not only the SA of but also the CA, CR, FA and FR (in %) (note the the $CA + FR = 100$ %; and $CR + FA = 100$ %).

Table 10.6: SA, CA, CR, FA and FR obtained in test d by the MLP trained with 6 nodes in the hidden layers.

	SA	CA (%)	CR (%)	FA (%)	FR (%)
Vowels	82.36	82.31	100.00	0.00	17.69
Fricatives	69.57	68.00	84.62	15.38	32.00
Affricates	52.69	49.40	80.00	20.00	50.60
Voiced plosives	70.67	70.67	—	—	29.33
Unvoiced plosives	52.43	52.43	—	—	47.57
Nasals	78.26	79.65	0.00	100.00	20.35
Palatals	73.33	78.57	0.00	100.00	21.43
Liquids	69.47	69.15	100.00	0.00	30.85

This table has to be analysed carefully, since the number of correctly and incorrectly pronounced elements is highly unbalanced. For example, there are no incorrectly pronounced phones among the plosives, so that SA and CA coincide, and this does not give us a clue about how the classification of incorrectly pronounced phones behave. On the other hand, we have very few incorrectly pronounced phones for nasals (2), palatals (1) and liquids (1), so that a bad classification of a such a phone can lead to a CR of 100 % or FA of 100 %.

Apart from this, the results are in accordance with the figures of section 10.3.2. Vowels are the group where correctly and incorrectly pronounced instances are best discerned. It is the group that obtains the best CA, and also a CR of 100 % (for 2 incorrectly pronounced instances). Palatals, affricates and fricatives are the groups that appear most overlapped in the figures, and the corresponding results are not very good. The most surprising result is the CA of unvoiced plosives; according to the parameter histogram, a better result was expected for this phone group.

10.4.4 Conclusions

In this section, different MLPs have been trained using different parameter sets, in order to see whether one parameter is more efficient than others at the time of deciding whether a phoneme has been correctly or incorrectly pronounced. Overall, using 6 nodes in the hidden layer of the MLP, the most discriminative parameter is the GOP score. Besides, previous and posterior phones do not seem to provide useful information, neither for GOPs, nor for the rest of parameters.

Using 64 nodes in the hidden layer instead, not all the best results per group are obtained using just GOP scores (group d). For example, liquids obtain the total best result for group b, and nasals obtain the total best result for groups c and e. Although it could be expected that more nodes in the hidden layer of the MLP or even more layers could be beneficial for the results, it has been proved that generalisation is better performed using few nodes.

10.5 Conclusions

In this chapter:

- We have given a definition of what a correctly and incorrectly pronounced phones are, from a practical point of view of acoustic signal processing.
- We have analysed different parameters in order to see which of them provide useful information at the time of training a Neural Network (in this case an MLP) to be used as a classifier.
- We have seen that GOP scores are —among the parameters analysed in this section— the most efficient parameters. The rest of the parameters introduce noise in more or less extent.
- Generally speaking, it seems better to use 6 nodes in the hidden layer of the MLP than 64.
- Results are quite coherent with the histograms of the parameters of correctly and incorrectly pronounced phones, except for unvoiced plosives, which present much lower results than expected.
- These results have been obtained for cases where a binary classification is needed. Using an intermediate category could make results be more useful.

PART IV

Summary and outlook

CHAPTER 11

Summary and outlook

11.1 Thesis contributions

This thesis has studied two ASR-based CALL applications for Basque: CAPT and WWSV. To that end, a standard ASR acoustic database has been used, since at the moment no Basque speech database suitable for CALL purposes exists. Both strategies rely on phone verification techniques, which means that the system has to automatically make decisions about when a phoneme is correctly or incorrectly pronounced. A novel technique has been proposed to cope with this issue: a population of incorrectly pronounced phones is obtained simulating localised errors (artificial errors), so that the GOP distributions of both the correctly and incorrectly pronounced phones are obtained. Thus, EER points can be obtained for each GOP distribution pair, to be used as thresholds. Shifted points could also be obtained to adjust the system to the level of the student.

GOP thresholds have been obtained using different HMMs: for CAPT, only native speakers of the *Basque Speecon-like* database were chosen to train the HMMs, since reference phoneme realisations are needed to compare the phones uttered by the students. However, for WWSV, speakers with "low level" skills in Basque have also been included in the training. This is due to the fact that the system will process the non-native students' speech when they are solving grammar exercises orally.

The on-line implementation has also been an important issue in this thesis. The system aims at being universally accessible via web, which offers two major advantages: on the one hand, the hardest processing is located in the server, and so the applications running in the user's device would be lightweight. On the other hand, the implementation in HTML5 makes the system cross-platform.

The server implementation presents several drawbacks, since the input audio will be differently recorded by each user, using a different device, in a different environment. To cope with this issue, a novel on-line CMVN technique has been introduced, based on a method proposed in this work: the Multi-Normalisation Scoring (MNS). The proposed CMVN technique allows normalising the MFCCs of a new incoming signal frame by frame, with no delay, obtaining promising results. Additionally, the MNS method has also been used to create a new on-line VAD system, which has proved to be very competitive compared with the state-of-the-art VAD systems, even in noisy conditions.

Additionally, a study of different parameter sets has been carried out to train NNs and substitute the GOP threshold method. All in all, GOP scores are, among the parameters considered, the most efficient parameters. The rest of the parameters introduce noise to a greater or lesser extent. Besides, the use of a small number of nodes in the hidden layer seems enough to train the NNs.

The overall contributions of this thesis can be summarised as follows:

- A thorough literature review of ASR-based CALL systems has been carried out. CAPT and SGP applications are the most popular nowadays. The use of GOP scores is a classical solution, which produces good results. Many works in the literature use students' mispronounced data to adapt acoustic models, but the development of specific databases to create CAPT applications is a hard task. Besides, they have the drawback that they cannot be used to develop tools for different L1 languages. The trend in the design of such CALL systems seems to be L1-independence. This is due to the fact that although it is easy to cover some specific usual errors of the speakers of a particular L1-L2 pair, the need for a more global system remains. All these issues are explained with detail in Chapter 2.
- The *Basque Speecon-like* database, the one used in this thesis, has been analysed in depth and adapted. The initial database contained all the present audio files but only part of the transcription files. So part of the annotation was done at the very beginning of this thesis. In addition, a phone inventory for Basque was established, in order to obtain a proper word lexicon to train acoustic models. Dialectal variations were also included in the lexicon to cope with the various dialectal variations in the database. The optimised version of the database has contributed to the development of the *SpeechTech4All* project, a project financed by the Spanish Ministry of Economy and Competitivity, focused on advanced research in all core speech technologies for all the official languages in Spain. This work and a description of the database are shown in Chapter 3.
- The *AhoSR* speech recognition system is described in Chapter 4. The first version of *AhoSR*, a simple but stable version, was built in 2012, and it was designed to manage simple word grammars. However, it has been gradually developed, and nowadays it can perform different tasks, such as phonetic recognition, word-grammar based recognition and large vocabulary speech recognition (LVCSR). Utterance verification techniques can also be used on these basic tasks, which were mainly devised for CAPT and SGP applications. By means of the on-line implementation, *AhoSR* has gained in flexibility, and it was used in different research projects, as in the final demo of *Ber2Tek*, a strategic research project

leaded by a consortium made up of different agents of the Basque Country. Moreover, it is currently implemented in the on-line dictionary of *Elhuyar*, a prestigious company of the Basque Country, whose main objective is to socialise Science and Technology and promote the progress of Basque language. *AhoSR* is the oral interface of the dictionary, which recognises input words and terms in Basque (currently in: https://hiztegiak.elhuyar.eus/).

- An in-depth study of the behaviour of HMMs has been carried out in Chapter 5. These acoustic models were trained from the *Basque Speecon-like* database, taking into account that it still contains a not properly labelled part, i.e. the free spontaneous speech part. Some direct conclusions have been drawn:
  - 32 Gaussians are enough for this database, since, from this point on, the phonetic error rate starts growing again.
  - The dictionary with alternative pronunciations created automatically does not provide with better acoustic models. Results are even worse when no manually corrected transcriptions are used, but they get closer when corrected data is included. This suggests that the benefits of using alternative pronunciations would be more noticeable the more correct the transcriptions are.
  - Using different data sets in different stages of the training process can improve the quality of the results. Initially using a manually corrected part along with the read subset and then the whole database can produce slight improvements.
  - Using CMVN better recognition results are obtained when analysing audio signals with mismatching channels, up to 30 % better. The recognition results testing audio signals recorded through the same audio channel are very similar to those obtained without cepstral normalisation.
- Chapter 6 describes the first evaluation of the WWSV-based SGP system, carried out with the first prototype of *AhoSR*. There, the novel strategy of obtaining two GOP score distributions (the corresponding to the correctly and incorrectly pronounced phones) was firstly tested, using the EER of both distributions as the decision threshold. The laboratory evaluations gave very good results, but tests in a more realistic environment were needed. The evaluation was carried out in two different Basque teaching institutions between Basque students of low level, giving as a result a Scoring Accuracy of 89.89 %. A short survey conducted among the students about the usefulness of the system and their overall impression showed that the user's perception of its performance was a little lower. Nevertheless, results were promising.
- The current trend in CALL systems is a client-server architecture on the Internet. Cross-platform audio capturing has created quite a major headache for the last several years, but nowadays the recent HTML5 and its *audio API* allows the browser to access the audio through any microphone connected to a computer. The WWSV-based SGP task needs to send audio on-the-fly. For that purpose, a

connection between the browser and the server is needed, which is not a priori possible within web technology. However, HTML5 defines *websockets* (included in the *web API*) which are able to set socket-like connections between the browser and the server. Combining these two APIs, an on-line audio sending system can be obtained by means of any browser that implements HTML5.

- The initial system had several improvements. One of them is a novel on-line VAD technique explained in Chapter 8. It is based on MNS, a method devised in this thesis which helps creating patterns that can be modelled. The overall results obtained by this new VAD system are very competitive compared to other state-of-the-art systems: Regarding the speech misclassification, similar results to the best tested VAD system are obtained; however, the silence misclassification rates are significantly better compared with the other systems. Moreover, results do not degrade very much with noise. This VAD was implemented in the final demo of the previously mentioned *Ber2Tek* project, as well as in the Elhuyar on-line dictionary. An article explaining the MNS-based on-line VAD has been published in a Q1 journal (`https://www.journals.elsevier.com/expert-systems-with-applications`). An off-line version of our MNS-based VAD can be tested in: `https://aholab.ehu.es/users/igor/VAD/index.php`.
- The web implementation of the WWSV-based SGP system requires an on-line MFCC normalisation method. Different strategies have been used in the literature to cope with this issue, and some of them have been tested in this thesis. The major problem is how to estimate the means and variances of cepstral parameters initially, without degrading the signal. The best results have been obtained by a *hybrid* approach, a novel method introduced here, which uses constant variance values previously computed from the database, and mean values calculated from the first N frames and updated recursively afterwards. Additionally, a novel and more robust on-line CMVN technique has also been proposed in this thesis. It is based in MNS as well, which means that it can be trained beforehand. Our MNS-based CMVN has no delay, because it does not depend on previous or future frames, and results are encouraging. The biggest drawback is that the method loses accuracy for noisy signals. All this is described in Chapter 9. An article is currently in preparation describing the MNS-based CMVN method introduced here for publication.
- DNNs have also been used to set thresholds for phoneme groups. Firstly, the concept of incorrectly pronounced phoneme has been revisited. Based on this idea, several DNNs have been trained using different parameter sets in order to see the impact of each parameter. The results of the experiments show that GOP scores are the most efficient parameters among durations and log-likelihoods of the previous, current and posterior phones. The results of the experiments are coherent with those obtained in the initial system.

11.2 Dissemination of results

Publications

The following paper is in preparation:

"On-line Cepstral Mean and Variance Normalisation (CMVN) based on Multi Normalisation Scoring (MNS)": to be sent to *IET Electronics Letters* journal.

The following papers have been published during the course of this research:

Journals:

1. **Igor Odriozola**, Inma Hernaez, Eva Navas: 'An on-line VAD based on Multi-Normalisation Scoring (MNS) of observation likelihoods'. *Expert Systems with Applications* (2018), vol. 110, pp. 52–61 (*JCR Impact Factor: 3.768*, **Q1**)
2. **Igor Odriozola**, Luis Serrano, Inma Hernaez, Eva Navas: 'The AhoSR Automatic Speech Recognition System'. *Advances in Speech and Language Technologies for Iberian Languages (Lecture Notes in Computer Science)* (2014), vol. 8854, pp. 279–288.
3. **Igor Odriozola**, Oliver Jokisch, Inma Hernaez, Rüdiger Hoffmann: 'Diseño y desarrollo de un sistema de evaluación automática de la pronunciación para el euskara'. *Procesamiento del Lenguaje Natural* (2012), vol. 49: pp. 101–108 (in Spanish).
4. **Igor Odriozola**, Eva Navas, Jon Sanchez, Inma Hernaez: 'Tratamiento léxico del euskara occidental basado en la división de radical y desinencia para reconocimiento de habla dialectal'. *Procesamiento del Lenguaje Natural* (2009), vol. 43: pp. 103–111 (in Spanish).
5. **Igor Odriozola**, Inma Hernaez, Eva Navas: 'Euskara eta Hizketa Teknologiak (Basque language and speech technologies)'. *BAT Journal of Sociolinguistics* (2008), vol. 66, pp. 125–133 (in Basque).

Contributions in conferences directly related to the research of this thesis:

1. **Igor Odriozola**, Inma Hernaez, Eva Navas, Luis Serrano, Jon Sanchez: 'The observation likelihood of silence: analysis and prospects for VAD applications'. *Proc. of IberSPEECH (ISCA)* (2018), pp. 50–54, Barcelona (Spain).
2. **Igor Odriozola**, Inma Hernaez, M. Ines Torres, L. Javier Rodriguez-fuentes, Mikel Penagarikano, Eva Navas: 'Basque Speecon-like and Basque SpeechDat MDB-600: speech databases for the development of ASR technology for Basque'. *Proc. of International Conference on Language Resources and Evaluation (LREC) (ELRA)* (2014), pp. 2658—2665, Reykjavik (Iceland).

3. **Igor Odriozola**, Oliver Jokisch, Inma Hernaez, Rüdiger Hoffmann: 'A pronunciation tutoring system for Basque - First development steps'. *Elektronische Sprachsignalverarbeitung (ESSV)* (2012), pp. 101–108, Cottbus (Germany).
4. **Igor Odriozola**, Eva Navas, Inma Hernaez, Iñaki Sainz, Ibon Saratxaga, Jon Sanchez, Daniel Erro: 'Using an ASR database to design a pronunciation evaluation system in Basque'. *International Conference on Language Resources and Evaluation (LREC) (ELRA)* (2012), pp. 4122–4126, Istanbul (Turkey).
5. **Igor Odriozola**, Inma Hernáez, Eva Navas: 'Design of a message verification tool to be implemented in CALL systems'. *Proc. of IberSPEECH (ISCA)* (2012), pp. 251–259, Madrid (Spain).
6. Igor Leturia, Arantza del Pozo, David Oyarzun, Urtza Iturraspe, Xabier Arregi, Kepa Sarasola, Arantza D. de Ilarraza, Eva Navas, **Igor Odriozola**, Iñaki Sainz: 'Web Communication Protocols for Coordinating the Modules of AnHitz, a Basque-Speaking Virtual 3D Expert on Science and Technology'. *Proc. of Web Services and Processing Pipelines in HLT workshop (LREC workshop) (ELRA)* (2010), pp. 60–67, Valletta (Malta).
7. Iker Luengo, Eva Navas, **Igor Odriozola**, Inma Hernaez, Iñaki Sainz, Daniel Erro: 'Modified LTSE-VAD Algorithm for Applications Requiring Reduced Silence Frame Misclassification'. *Proc. of International Conference on Language Resources and Evaluation (LREC) (ELRA)* (2010), pp. 1539–1544, Valletta (Malta).
8. Ibon Saratxaga, Inma Hernaez, **Igor Odriozola**, Eva Navas, Iker Luengo, Daniel Erro: 'Using Harmonic Phase Information to Improve ASR Rate'. *Interspeech (ISCA).* (2010), pp. 1185–1188, Makuhari (Japan).
9. Igor Leturia, Arantza del Pozo, Kutz Arrieta, Urtza Iturraspe, Kepa Sarasola, Arantza D. de Ilarraza, Eva Navas, **Igor Odriozola**: 'Development and Evaluation of AnHitz, a Prototype of a Basque-Speaking Virtual 3D Expert on Science and Technology'. *International Multiconference on Computer Science and Information Technology (IMCSIT).* (2009), pp. 235–242, Mragowo (Poland).
10. Gotzon Aurrekoetxea, Jon Sanchez, **Igor Odriozola**: 'EDAK: A corpus to analyse linguistic variations'. *Proc. of A Survey on Corpus-based Research / Panorama de investigaciones basadas en corpus (AELINCO)* (2009), pp. 489–503, Murcia (Spain).

Other publications

Other works related to speech technologies where I have participated in:

1. Inma Hernaez, Eva Navas, **Igor Odriozola**, Kepa Sarasola, Arantza D. de Ilarraza, Igor Leturia, Beñat Oihartzabal, Jasone Salaberria:

‘The Basque Language in the Digital Age – Euskara Aro Digitalean’. *META-NET White Paper Series* (2012).

2. Iñaki Sainz, Daniel Erro, Eva Navas, Inma Hernaez, Jon Sánchez, Ibon Saratxaga, **Igor Odriozola**: ‘Versatile Speech Databases for High Quality Synthesis for Basque’. *Proc. of International Conference on Language Resources and Evaluation (LREC) (ELRA).* (2012) pp. 3308–3312, Istanbul (Turkey).
3. Ibon Saratxaga, Inma Hernaez, Eva Navas, Iñaki Sainz, Iker Luengo, Jon Sanchez, **Igor Odriozola**, Daniel Erro: ‘AhoTransf: A Tool for Multiband Excitation Based Speech Analysis and Modification’. *Proc. of International Conference on Language Resources and Evaluation (LREC) (ELRA)* (2010), pp. 3732–3737, Valletta (Malta).
4. Iñaki Sainz, Daniel Erro, Eva Navas, Inma Hernaez, Jon Sanchez, Ibon Saratxaga, **Igor Odriozola**, Iker Luengo: ‘Aholab Speech Synthesizers for Albayzin 2010’. *Proc. of VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop - FALA* (2010), pp. 343–347, Vigo (Spain).
5. Daniel Erro, Iñaki Sainz, Iker Luengo, **Igor Odriozola**, Jon Sanchez, Ibon Saratxaga, Eva Navas, Inma Hernaez: ‘HMM-based Speech Synthesis in Basque Language using HTS’. *Proc. of VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop - FALA* (2010), pp. 67–70, Vigo (Spain).
6. Iñaki Sainz, Daniel Erro, Eva Navas, Inma Hernaez, Ibon Saratxaga, Iker Luengo, **Igor Odriozola**: ‘The AHOLAB Blizzard Challenge 2009 Entry’. *Blizzard Challenge 2009 (http://festvox.org/blizzard/blizzard2009.html).* (2009), Edinburgh (UK).
7. Iker Luengo, Eva Navas, Iñaki Sainz, Ibon Saratxaga, Jon Sanchez, **Igor Odriozola**, J.J. Igarza, Inma Hernaez: ‘Grabación de una Base de datos Bilingüe Euskera/Castellano para Verificación de Locutor’. *Proc. of V Jornadas en Tecnología del Habla* (2008), pp. 195–198, Bilbao (Basque Country).
8. Ibon Saratxaga, Eva Navas, Inma Hernaez, Jon Sanchez, Iker Luengo, **Igor Odriozola**, Eneritz de Bilbao: ‘Evaluación Subjetiva de una Base de Datos de Habla Emocional para Euskera’. *Proc. of V Jornadas en Tecnología del Habla* (2008), pp. 191–194, Bilbao (Basque Country).
9. Iñaki Sainz, Inma Hernaez, Eva Navas, Jon Sanchez, Iker Luengo, Ibon Saratxaga, **Igor Odriozola**, Eneritz de Bilbao, Daniel Erro: ‘Descripción del Conversor de Texto a Voz AhoTTS Presentado a la Evaluación Albayzin TTS 2008’. *Proc. of V Jornadas en Tecnología del Habla* (2008), pp. 96–99, Bilbao (Basque Country).
10. Iñaki Sainz, Ibon Saratxaga, Eva Navas, Inma Hernaez, Jon Sanchez, Iker Luengo, **Igor Odriozola**: ‘Subjective Evaluation of an Emotional Speech

Database for Basque'. *Proc. of International Conference on Language Resources and Evaluation (LREC) (ELRA)* (2008), paper 437, Marrakech (Morocco).

11. Iker Luengo, Eva Navas, Iñaki Sainz, Ibon Saratxaga, Jon Sanchez, **Igor Odriozola**, Inma Hernaez: 'Text independent speaker identification in multilingual environments'. *Proc. of International Conference on Language Resources and Evaluation (LREC) (ELRA)* (2008), paper 461, Marrakech (Morocco).
12. Iker Luengo, Eva Navas, Iñaki Sainz, Ibon Saratxaga, Jon Sanchez, **Igor Odriozola**, J.J. Igarza, Inma Hernaez: 'Building a Basque/Spanish bilingual database for speaker verification'. *Workshop Collaboration: interoperability between people in the creation of language resources for less-resourced languages (SALTMIL)* (2008), pp. 23–26, Marrakech (Morocco).
13. Iker Luengo, Eva Navas, Iñaki Sainz, Ibon Saratxaga, Jon Sanchez, **Igor Odriozola**, Inma Hernaez: 'Identificación de locutores en entornos multilingües'. *Proc. of IV Jornadas en Reconocimiento Biométrico de Personas* (2008) pp. 133-140, Valladolid (Spain).

11.3 Future work

Currently, we consider that the technology to implement a Basque CAPT system and a WWSV-based SGP system in a remote server is ready. The systems introduced in this work have met their initial expectations, and we have proposed solutions for the problems that arise when they are implemented in a server, as the on-line cepstral normalisation.

Based on the technology we have developed, a project is being defined to implement the CAPT system in *ikasbil* (`www.ikasbil.eus`), the Basque learning website of HABE (Institution for Adults Alphabetisation and Revasconisation). HABE is a Basque government institution which works to help adults learn the Basque language, by means of 107 euskaltegis (Basque language education centres) and about 35 000 students (in the 2016/2017 school year). The 12.24 % of the students has opted for self-learning. Additionally, between 3000 and 4000 people outside of the Basque Country study Basque through HABE each year, by means of ikasbil. The aim of the project is to implement the current CAPT system as an additional tool, and step by step make the necessary improvements until a robust CAPT system is achieved. In that system, not only phoneme realisations will be assessed, but also prosody.

A necessary future work will be the integration of Neural Networks in the speech recogniser system. As the first step, acoustic models trained with DNNs must be integrated, not only to obtain better recognition results, but also to get more efficient GOPs (as explained in [66]). The analysis of meaningful parameters and the use of DNNs to classify uttered phones is also a future way to explore. This would do the system more robust, easy to use and universal.

APPENDIX A

Configurable parameters of AhoSR

This appendix contains a summary of the most important *AhoSR* parameters and the values which may currently be assigned to them. The parameter/value pairs can be included in a configuration file, which override default values at the time when the application is loading. The values marked with an asterisk (*) indicate the default value of a parameter.

Other parameters, such as those related to file management and data storing, have been omitted, as they are not relevant. Theses parameters are mainly used to indicate the location of different resources, as well as the format of the output data such as results, audio or MFCC files.

A.1 General parameters

Parameters related to the general operation mode and audio input mode.

Table A.1: General configurable parameters in *AhoSR*

Parameter	Value	Description
GENERAL		
OPERATION MODE	*RECOGNITION**	Performs the recognition decoding process.
	MFCC	Extracts MFCCs of the input signal and stores them in a file.
AUDIO MODE	*WAV_AUDIO**	Wav format files.
	DIRECT_AUDIO	Direct audio from microphone.
	SOCKET_RAW_-AUDIO	Raw PCM audio data through a socket connection.

A.2 Input audio

Parameters related to the quality of the input audio signal.

Table A.2: Input audio quality related configurable parameters in *AhoSR*

Parameter	**Value**	**Description**
INPUT AUDIO		
SAMPLE RATE	*16000**	Sample rate at which the incoming audio signal is processed.
BITS PER SAMPLE	*16**	Number of bits used to quantify audio samples.

A.3 MFCC extraction parameters

Parameters related to the extraction and characteristics of MFCCs.

Table A.3: MFCC extraction related configurable parameters in *AhoSR*

Parameter	**Value**	**Description**
MFCC PARAMETERS		
FRAME RATE	*10**	The shift of the analysis window (in ms).
FRAME LENGTH	*25**	The length of the analysis window (in ms).
WINDOW TYPE	*HAMMING**	Hamming window.
	RECT	Rectangular window.
	BART	Bartlett window.
	HANNING	Hanning window.
	BLACK	Blackman window.
MIN. FREQ. LIMIT	*0**	Lower limit for frequency analysis.
MAX. FREQ. LIMIT	*0**	Upper limit for frequency analysis.
NUMBER OF FILTERS	*26**	Number of MEL filters for the frequency analysis.
SCALE	*MEL**	Mel scale.
	BARK	Bark scale.

continued on next page . . .

Parameter	Value	Description
MFCC PARAMETERS		
NUMBER OF CEPSTRUM	*12**	Output number of Ceps coefficients.
C0	*1**	Includes 0th coefficient.
DELTA WINDOW	*2**	Window length for first derivatives $(X * 2 + 1)$.
ACC WINDOW	*2**	Window length for second derivatives $(X * 2 + 1)$.
LIFTERING	*22**	Liftering coefficient.
CMS	*false**	Apply CMS.
PREENF	*0.97**	Pre-emphasis coefficient.
HTK	*false**	To parametrise as in HTK.

A.4 Recognition task

Parameters related to the type of recognition task desired.

Table A.4: Recognition task related configurable parameters in *AhoSR*

Parameter	Value	Description
RECOGNITION TASK		
WORD NETWORK	*PHONETIC*	Phonetic recognition, using a structured triphone net.
	*WORD_GRAMMAR**	BNF and SLF grammar based recognition
	WORD_LOOP	Free word loop (choose LM for Continuous Speech Recognition).
	SENTENCE_VERIFICATION	Sentence verification (for CAPT purposes)
	ON-LINE_VERIFICATION	Sentence verification with word-by-word on-line feedback (for WWSV purposes).
	FORCED_ALIGNMENT	Forced alignment, both at word and phone level.
	MULTIPLE	Different grammars loaded and managed.

A.5 Language Modelling

Parameters related to the configuration of Language Modelling (LM).

Table A.5: Language Model related configurable parameters in *AhoSR*

Parameter	**Value**	**Description**
LANGUAGE MODELLING		
LM	*false**	Use language model probabilities.
LM MODE	*3-GRAM**	Definition of the highest N-gram order.

A.6 Search space organization

Parameters related to the organization of the nodes in search space.

Table A.6: Search space organization related configurable parameters in *AhoSR*

Parameter	**Value**	**Description**
SEARCH SPACE		
COMPRESSION TYPE	*NONE**	No compression is applied.
	PREFIX	Left-to-right node compression
	SUFFIX	PREFIX and right-to-left node compression.
UNIT EXPANSION	*true**	Word-end nodes are expanded to model coarticulation between words.
NON SPEECH EVENTS	*true**	Non speech events' HMMs are added in parallel to silence nodes.

A.7 HMMs

Parameters related to the topology of HMMs.

Table A.7: HMM topology related configurable parameters in *AhoSR*

Parameter	**Value**	**Description**
HMMs		
STATE NUMBER	*5**	State number of HMM (including end states).
GAUSSIAN NUMBER	*32**	Gaussian number of each GMM in each HMM state.

A.8 Pruning

Parameters related to different pruning techniques.

Table A.8: Pruning related configurable parameters in *AhoSR*

Parameter	Value	Description
PRUNING		
PT	*0**	Pruning Threshold of histogram pruning.
MAX PATHS	*0**	Maximum number of active tokens permitted at each instant.
GAUSSIAN SELECTION	*0**	No Gaussian selection is applied.
	1	PDE (Partial Distance Elimination)
TOKEN NUMBER	*1**	Number of tokens that each state can hold.

A.9 CMVN

Parameters related to Cepstral Mean and Variance Normalization (CMVN) technique.

Table A.9: CMVN related configurable parameters in *AhoSR*

Parameter	Value	Description
CMVN		
CMVN	*0**	No normalization is applied.
	1	Mean normalization.
	2	Mean and variance normalization.
CMVN ONLINE	*false**	On-line calculation of means (and variances).
CMVN UPDATING	*false**	Recursive updating of means (and variances).
CMVN INIT LOOK-AHEAD	*25**	Number of initial frames to estimate the initial values of means (and variances).
CMVN INIT PAST DATA	*NONE**	Name of the set with values of means (and variances) computed previously.
CMVN HYBRID	*false**	Hybrid approach (update means + past vars).
CMVN INIT VAD	*0**	Size (in frames) of the segment around the first non-speech-to-speech transition to be used to estimate initial values of means (and variances).

A.10 VAD

Parameters related to the configuration of the Voice Activity Detector (VAD).

Table A.10: VAD related configurable parameters in *AhoSR*

Parameter	**Value**	**Description**
VAD		
VAD	*NONE**	No VAD is used.
	EXTR	VAD based in energy.
	HMM SIL	VAD based on the central state GMM of the silence HMM.
MIN SPEECH FRAMES	*15**	Number of frames for a speech segment to be considered as speech.
MIN SIL FRAMES	*15**	Number of frames for a silence segment to be considered as silence.
SPEECH MARGIN	*0**	Number of silence frames added to a speech segment at both ends.

A.11 UV

Parameters related to the configuration of phone scoring (Utterance Verification).

Table A.11: UV related configurable parameters in *AhoSR*

Parameter	**Value**	**Description**
UV		
UV	*false**	Utterance Verification is applied over the Viterbi decoder paths.
UV UNIT	*WORD*	Word-level verification scores are computed.
	*PHONE**	Both word-level and phone-level verification scores are computed.

APPENDIX B

GOP, duration and log-likelihood histograms

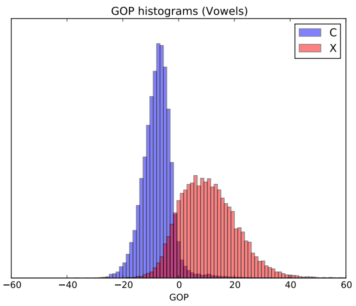
B.1 Histograms of correctly and incorrectly pronounced phonemes

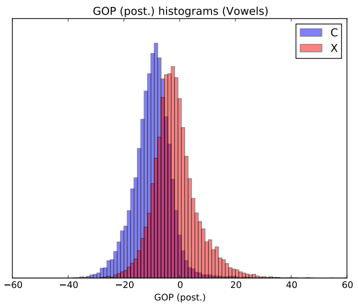
This section shows the GOP, durations and log-likehood histograms (and the ones of the previous and posterior phones) of correctly (C) and incorrectly (X) pronounced phones, broken down by their phone group, their duration and their position in a voice segment.

1. **VOWELS**

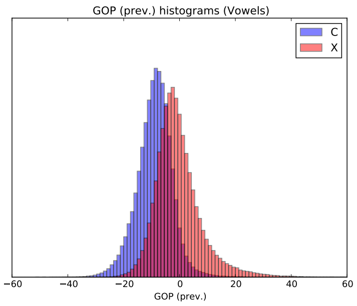
- GOP:

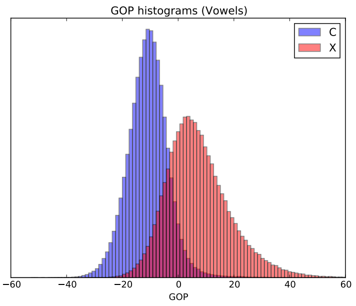
  – Left phone

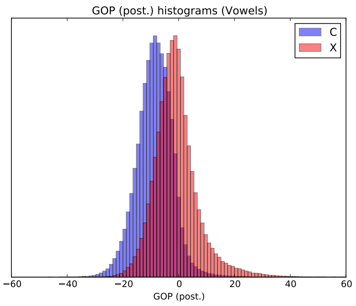

GOP histograms (Vowels)


GOP (post.) histograms (Vowels)

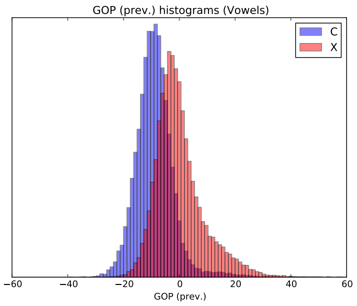
  – Middle phone:

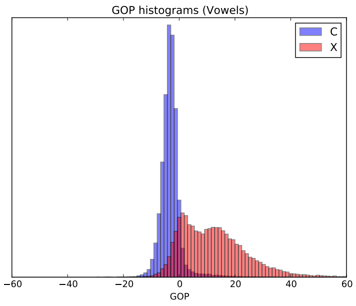

GOP (prev.) histograms (Vowels)


GOP histograms (Vowels)

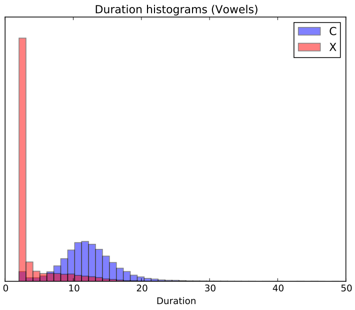

GOP (post.) histograms (Vowels)

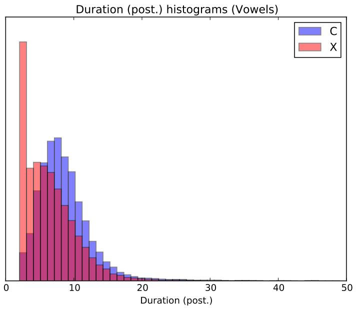
  – Right phone


GOP (prev.) histograms (Vowels)

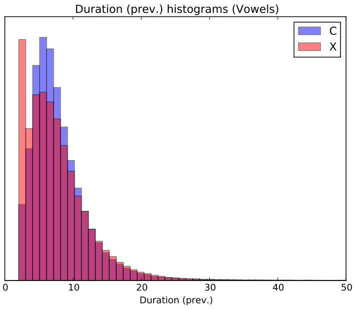

GOP histograms (Vowels)

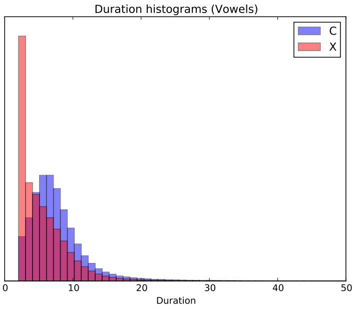
- Duration:
  - Left phone

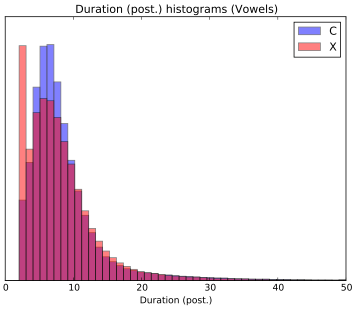




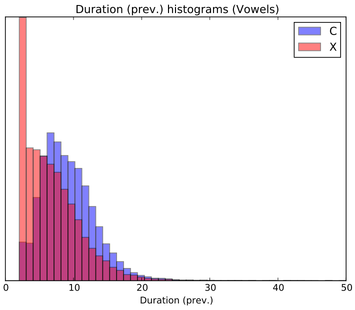
  - Middle phone:

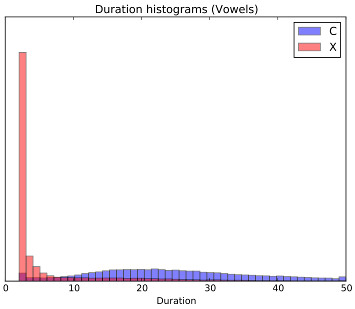




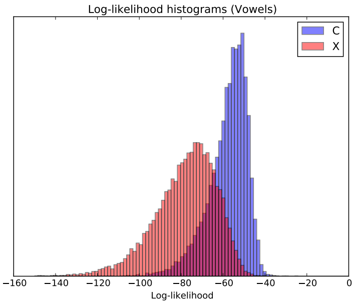


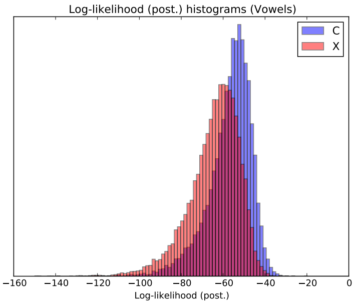
  - Right phone



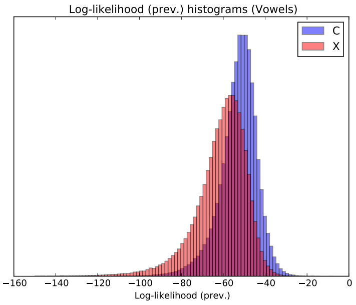


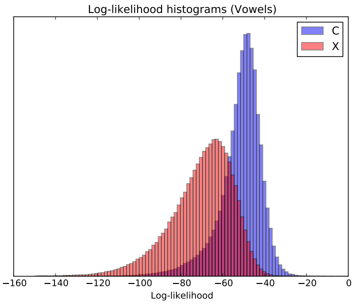
- Log-likelihood:
  - Left phone

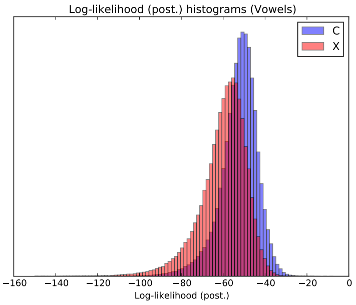




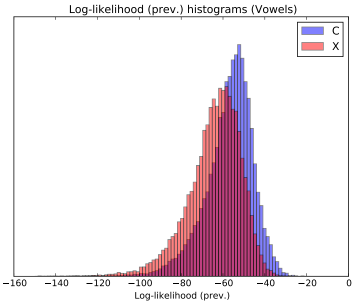
  - Middle phone:

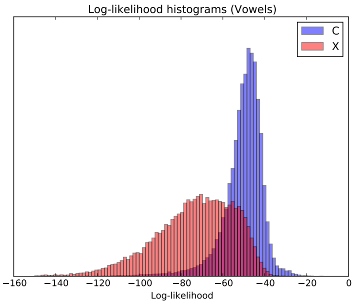






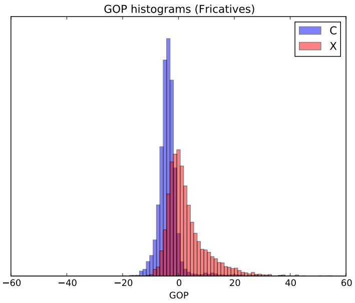
  - Right phone

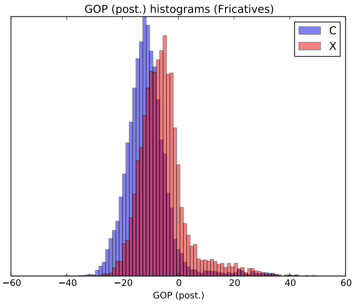




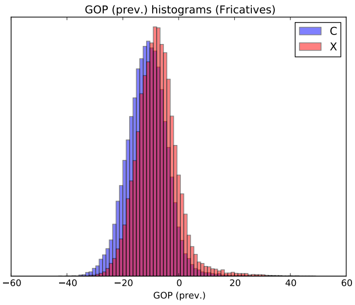
2. **FRICATIVES**

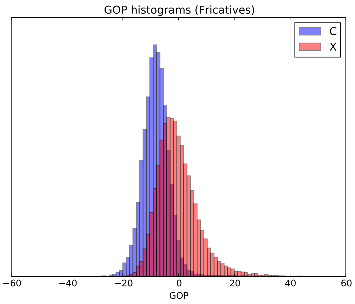
- GOP:
  - Left phone

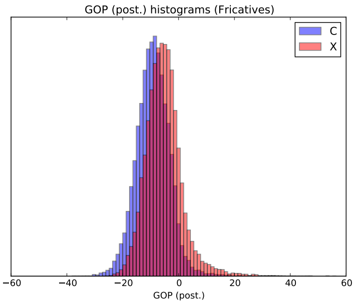




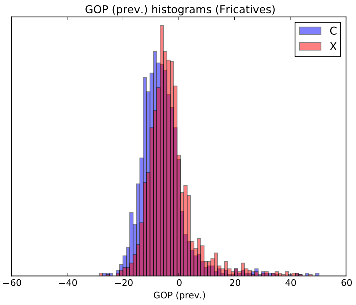
  - Middle phone:

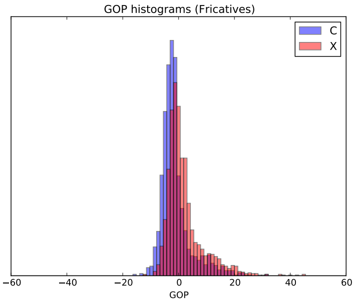




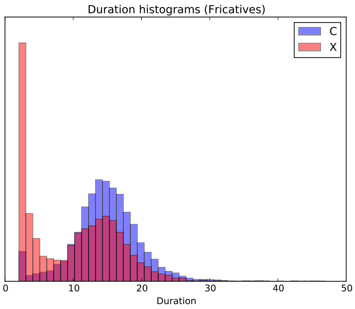


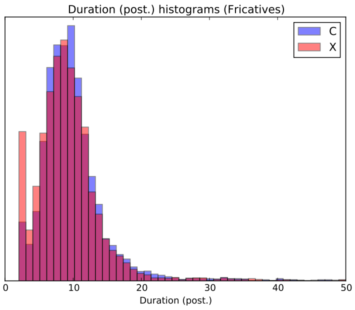
  - Right phone



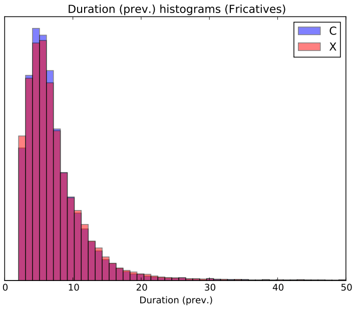


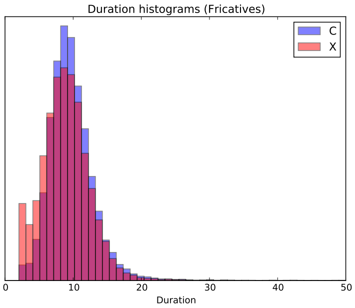
- Duration:
  - Left phone

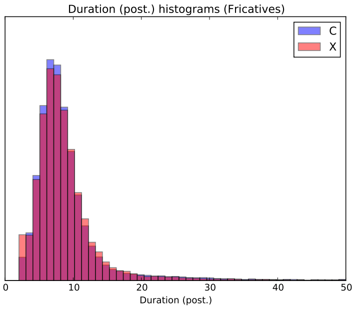

Duration histograms (Fricatives)


Duration (post.) histograms (Fricatives)

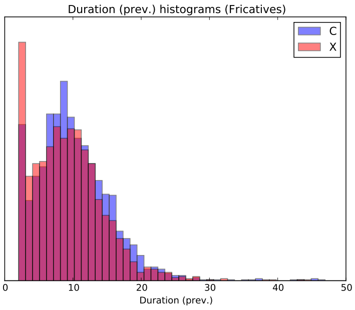
  - Middle phone:

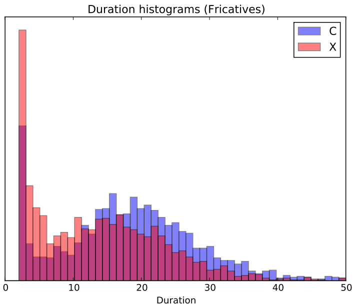

Duration (prev.) histograms (Fricatives)


Duration histograms (Fricatives)

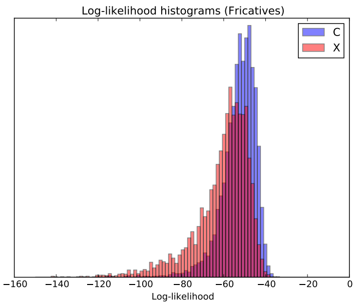

Duration (post.) histograms (Fricatives)

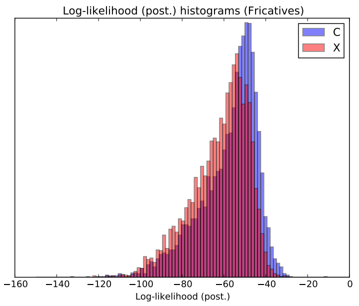
  - Right phone


Duration (prev.) histograms (Fricatives)

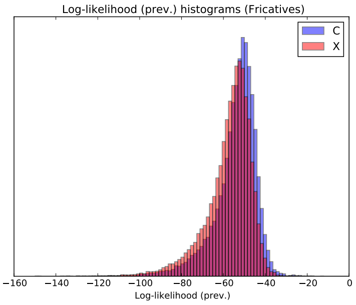

Duration histograms (Fricatives)

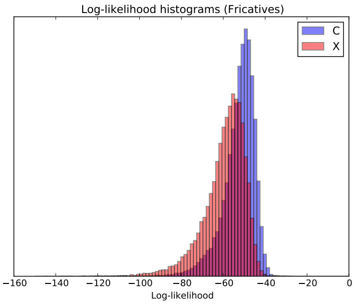
- Log-likelihood:
  - Left phone

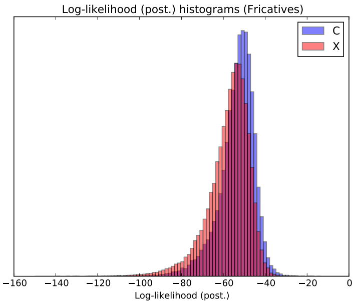




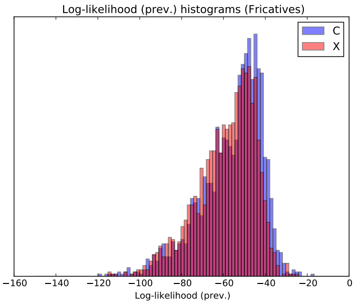
  - Middle phone:

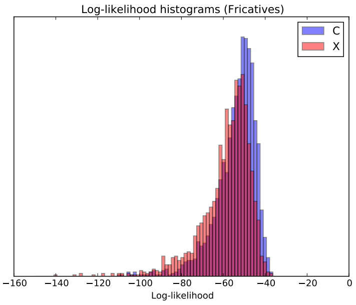






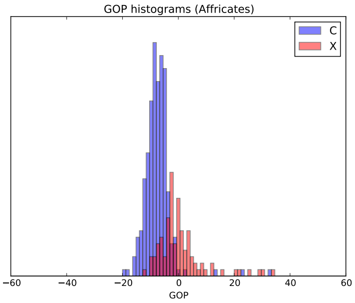
  - Right phone

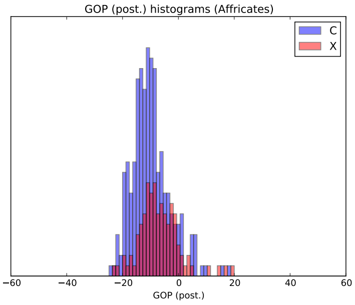




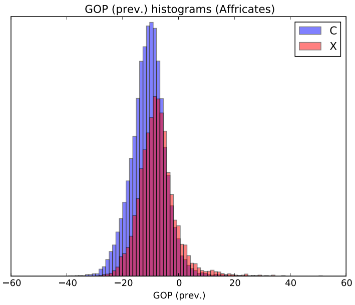
3. **AFFRICATES**

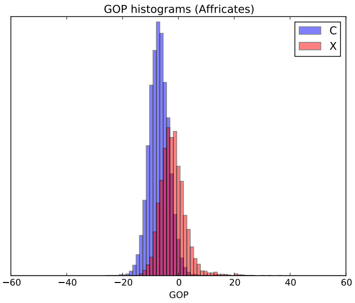
- GOP:
  - Left phone

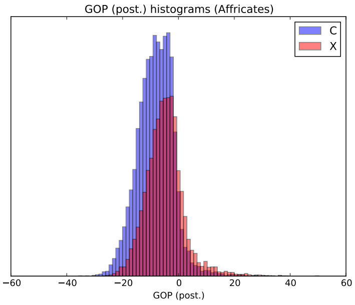




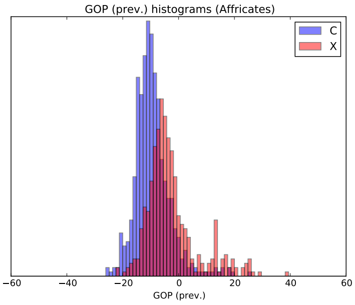
  - Middle phone:

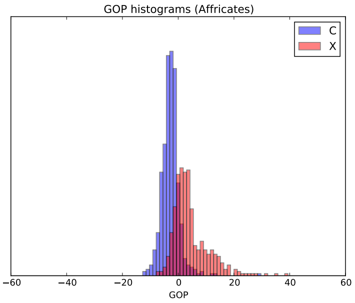




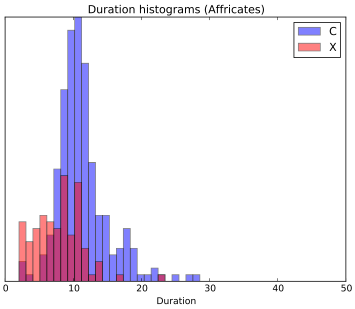


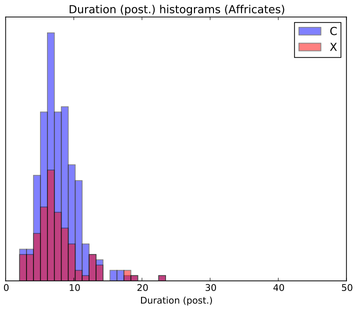
  - Right phone



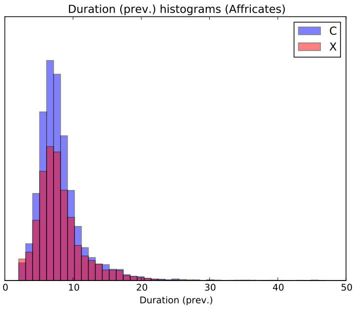


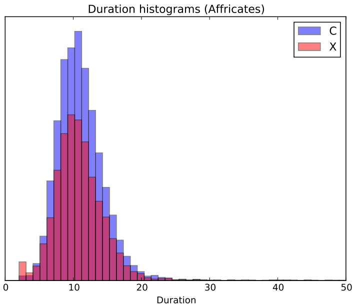
- Duration:
  - Left phone

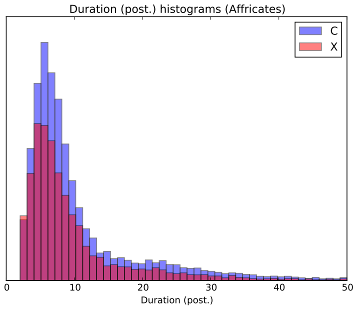




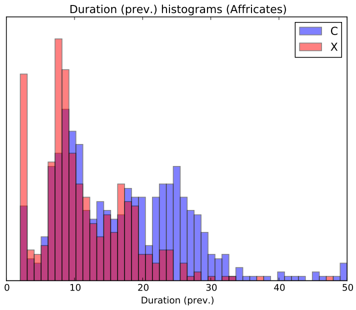
  - Middle phone:

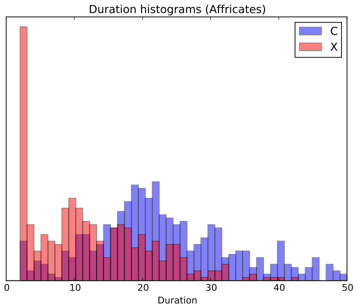




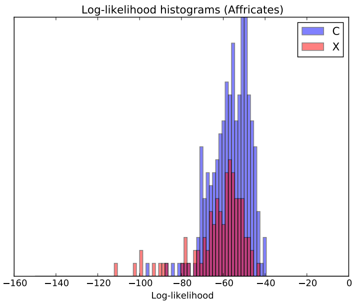


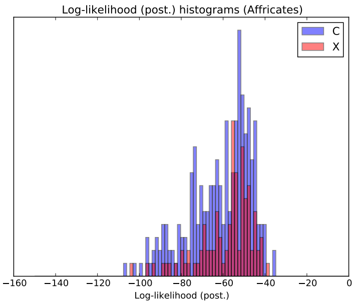
  - Right phone



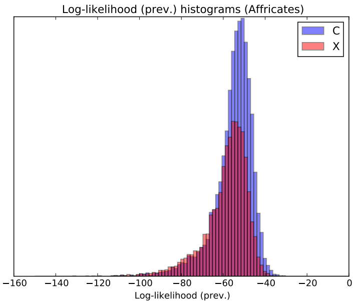


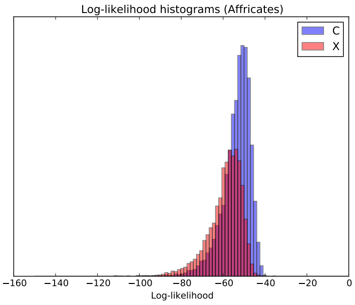
- Log-likelihood:
  - Left phone

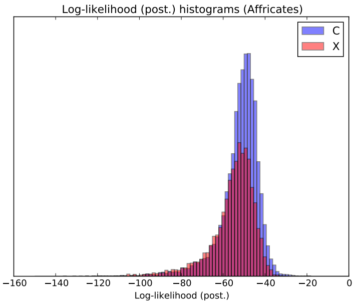




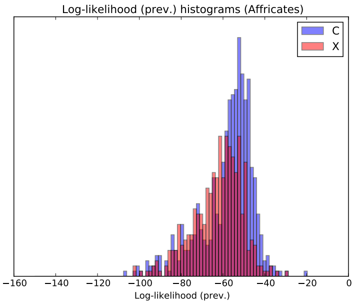
  - Middle phone:

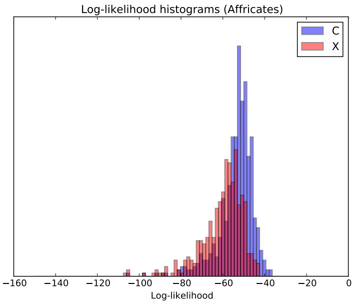






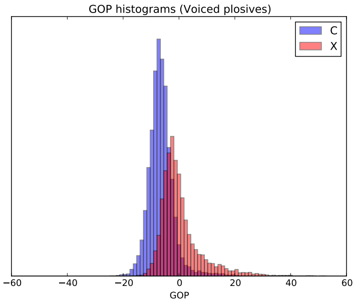
  - Right phone

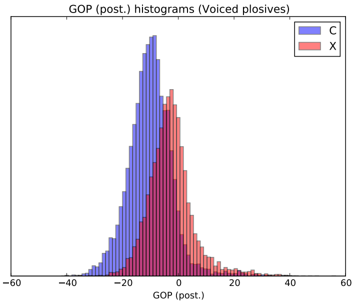




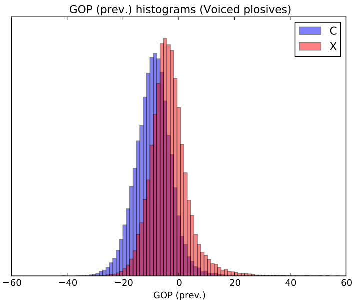
4. **VOICED PLOSIVES**

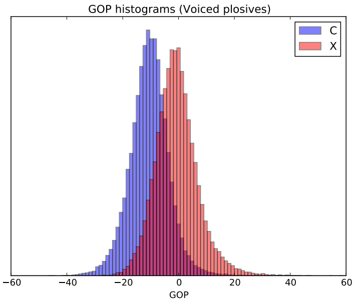
- GOP:
  - Left phone

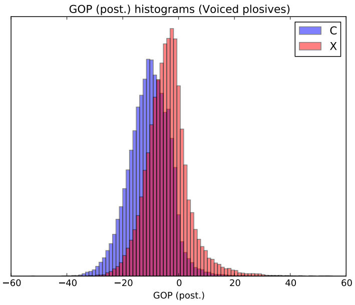




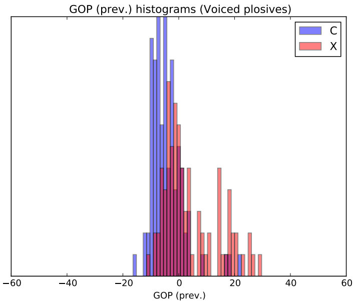
  - Middle phone:

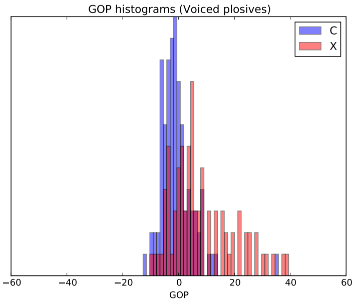




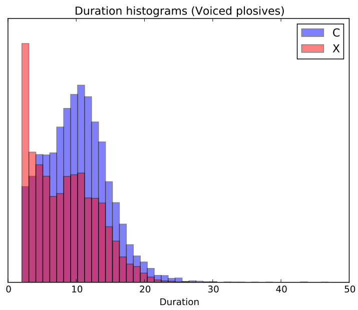


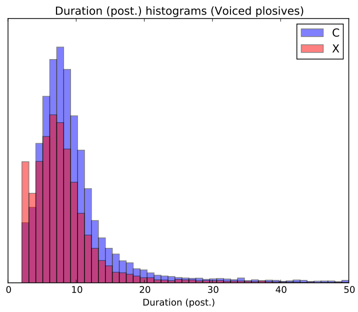
  - Right phone



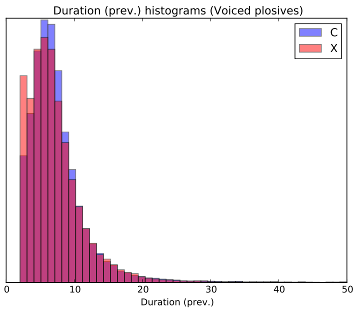


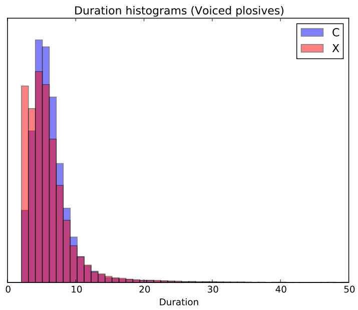
- Duration:
  - Left phone

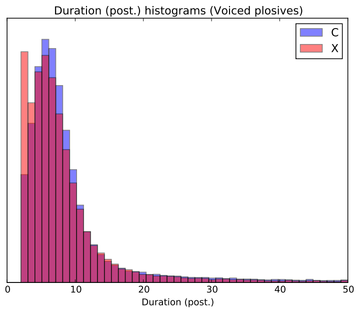




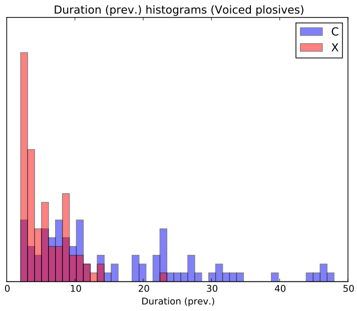
  - Middle phone:

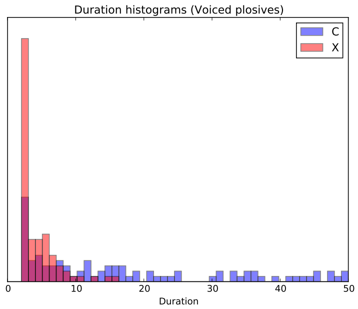




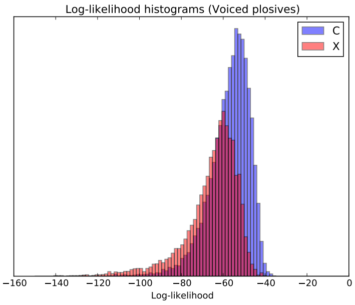


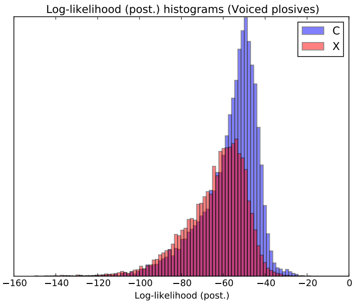
  - Right phone



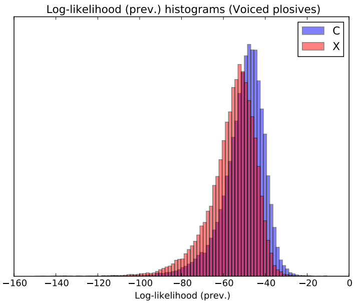


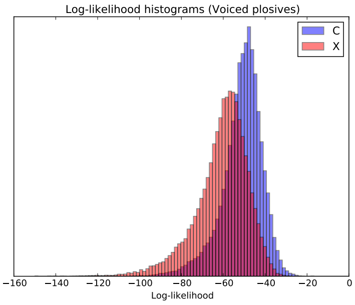
- Log-likelihood:
  - Left phone

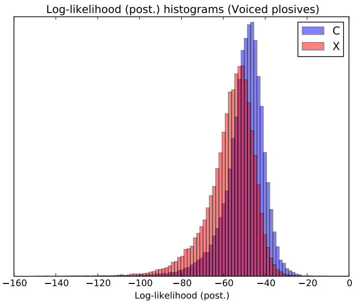




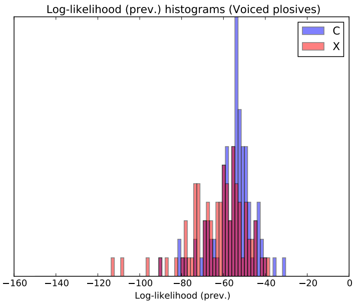
  - Middle phone:

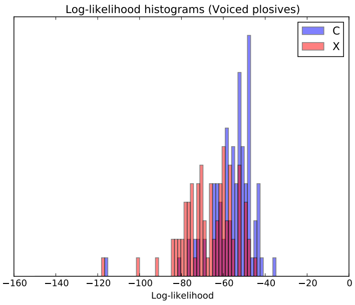






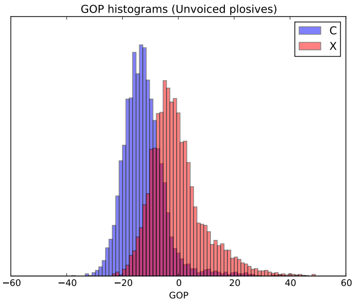
  - Right phone

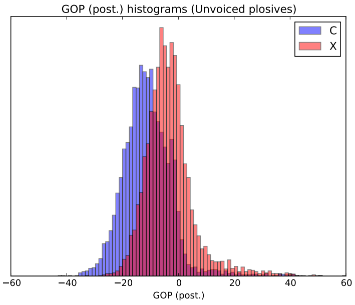




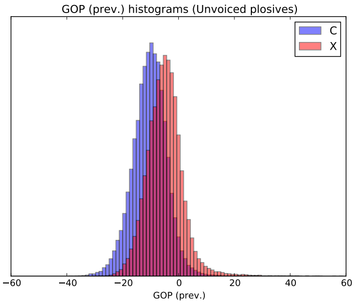
5. **UNVOICED PLOSIVES**

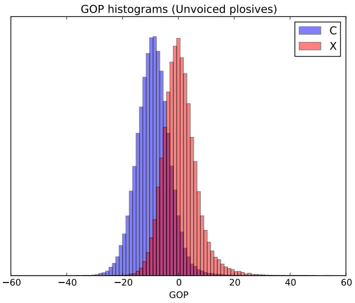
- GOP:
  - Left phone

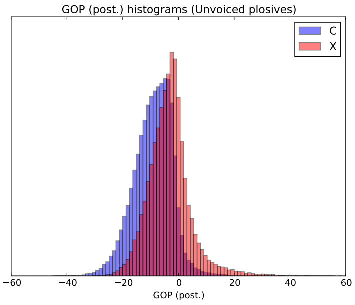




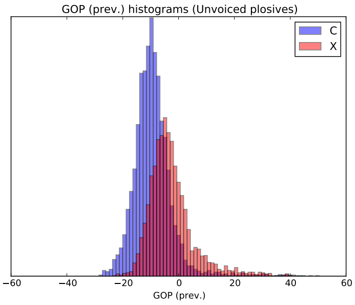
  - Middle phone:

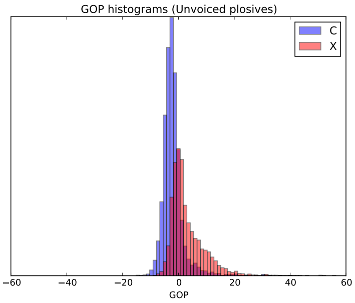




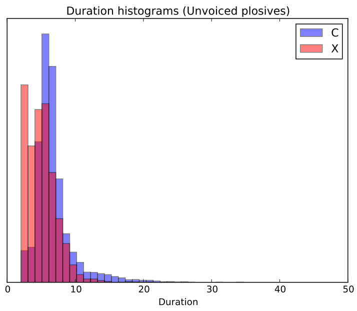


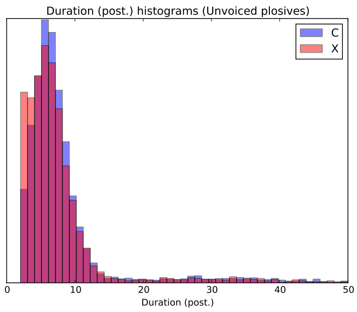
  - Right phone



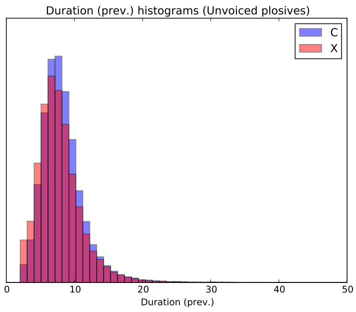


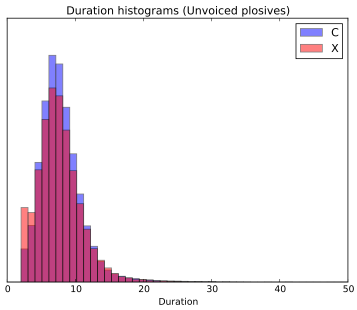
- Duration:
  - Left phone

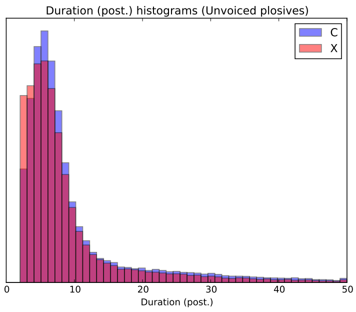




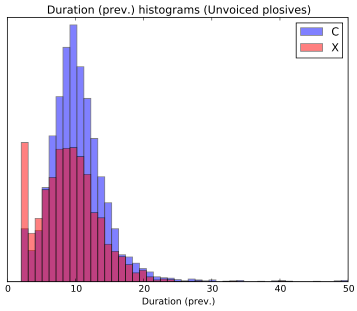
  - Middle phone:

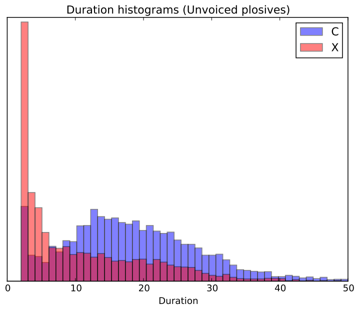




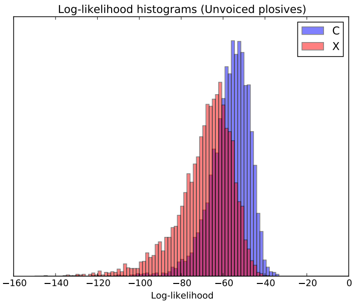


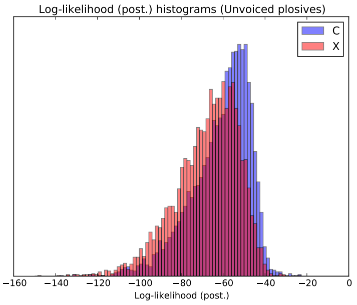
  - Right phone



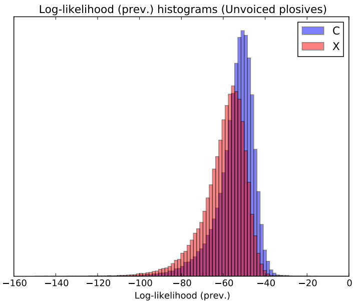


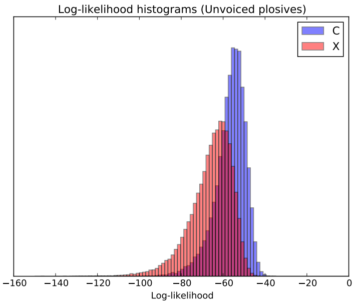
- Log-likelihood:
  - Left phone

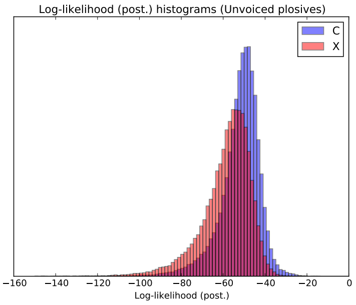

Log-likelihood histograms (Unvoiced plosives)


Log-likelihood (post.) histograms (Unvoiced plosives)

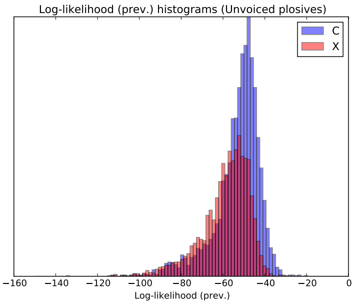
  - Middle phone:

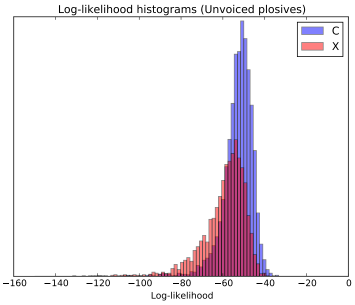

Log-likelihood (prev.) histograms (Unvoiced plosives)


Log-likelihood histograms (Unvoiced plosives)


Log-likelihood (post.) histograms (Unvoiced plosives)

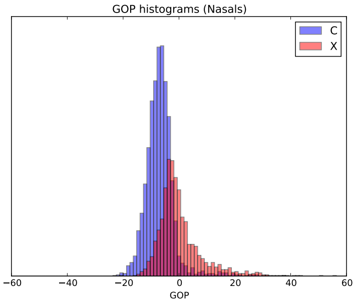
  - Right phone

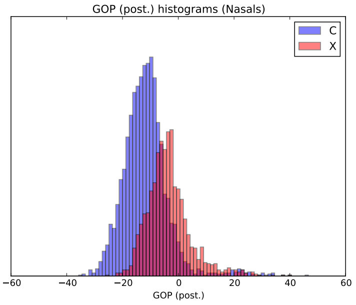

Log-likelihood (prev.) histograms (Unvoiced plosives)


Log-likelihood histograms (Unvoiced plosives)

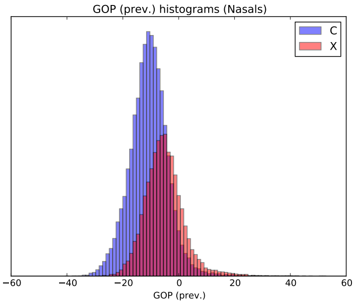
6. NASALS

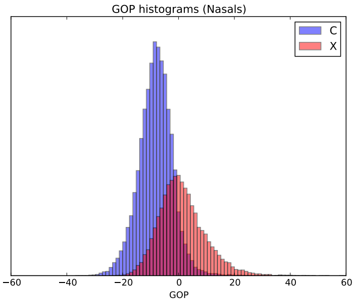
- GOP:
  - Left phone

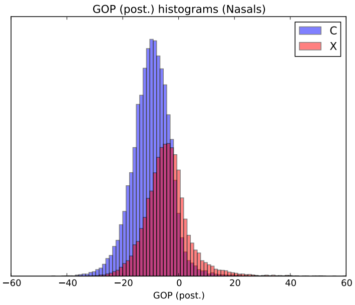




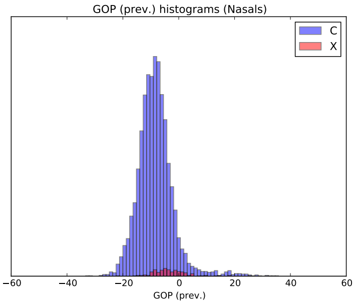
  - Middle phone:

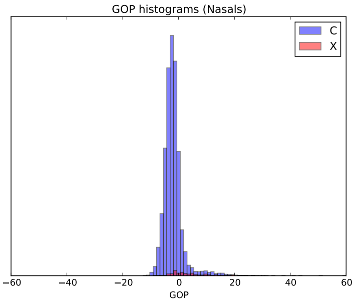




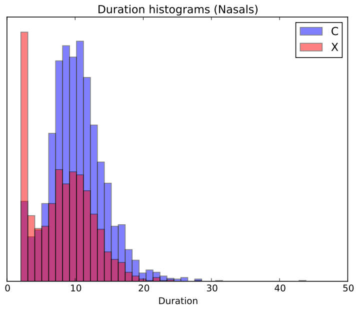


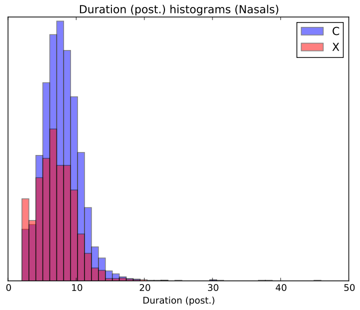
  - Right phone



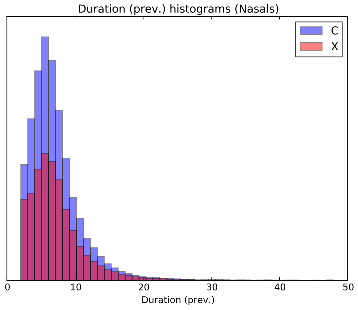


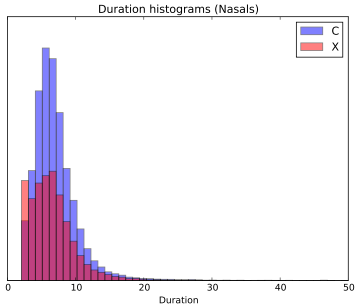
- Duration:
  - Left phone

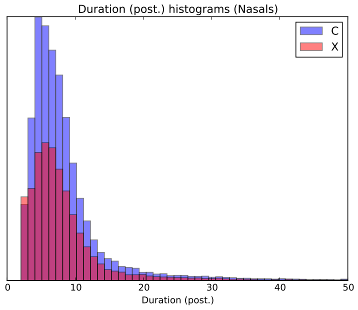

Duration histograms (Nasals)


Duration (post.) histograms (Nasals)

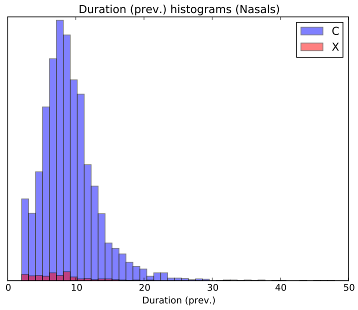
  - Middle phone:

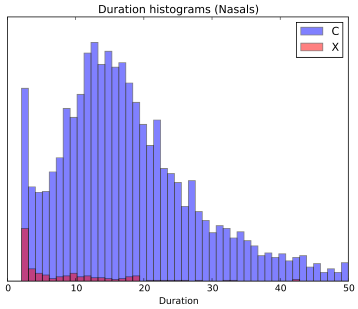

Duration (prev.) histograms (Nasals)


Duration histograms (Nasals)

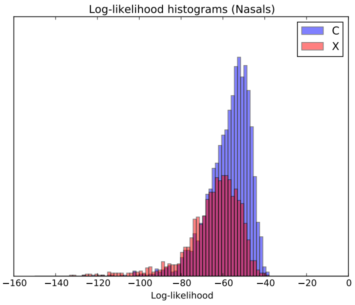

Duration (post.) histograms (Nasals)

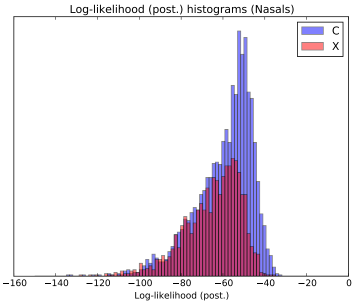
  - Right phone


Duration (prev.) histograms (Nasals)

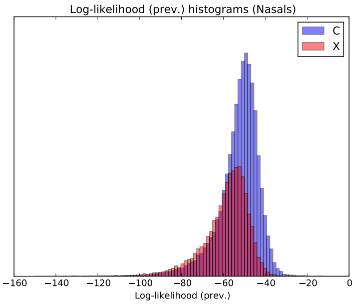

Duration histograms (Nasals)

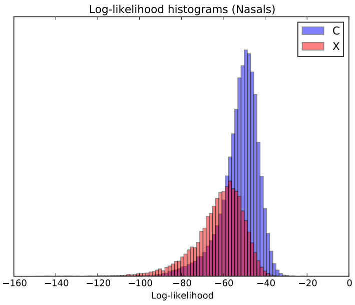
- Log-likelihood:
  - Left phone

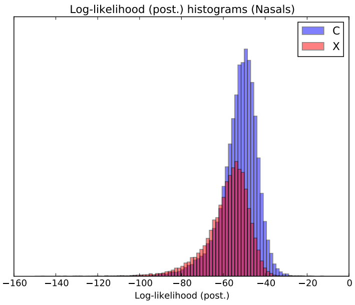




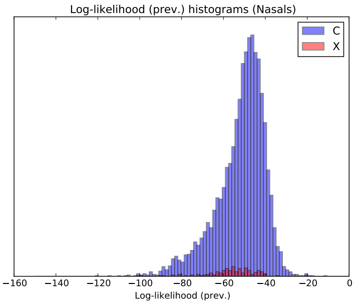
  - Middle phone:

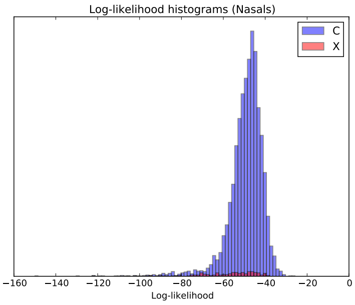






  - Right phone

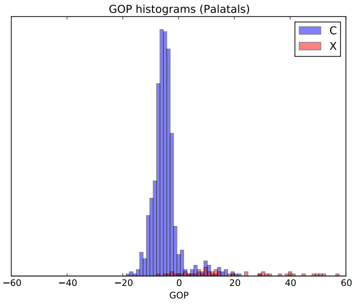


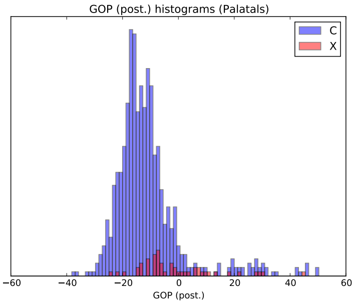


7. **PALATALS**

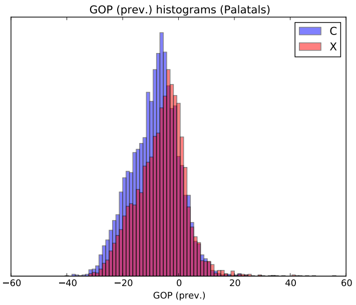
- GOP:

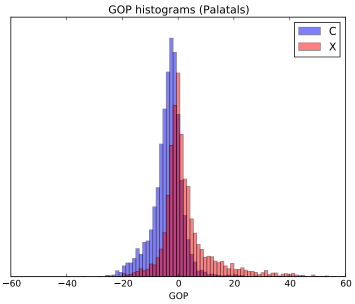
– Left phone

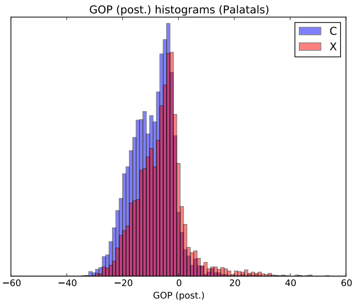




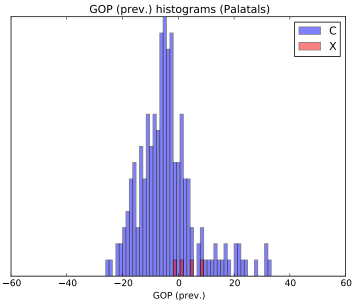
– Middle phone:

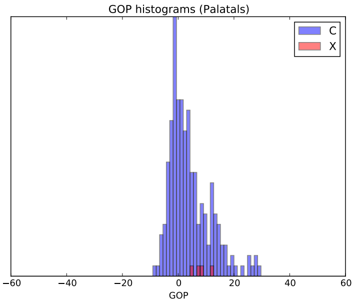




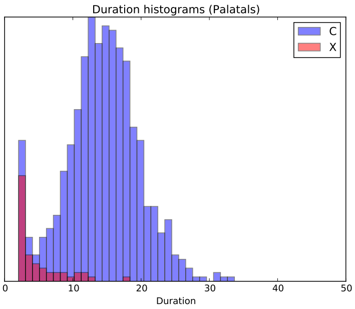


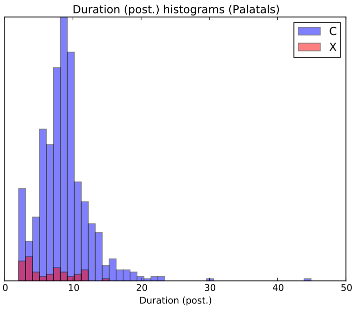
– Right phone



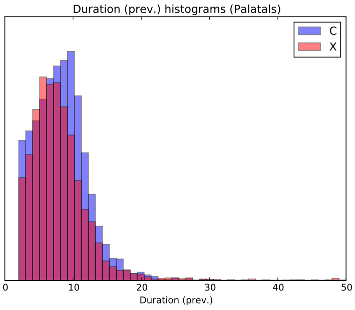


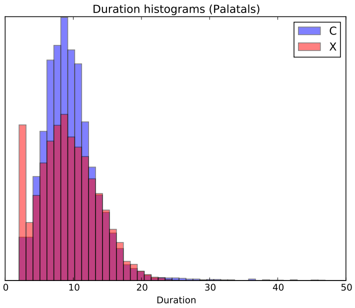
- Duration:
  - Left phone

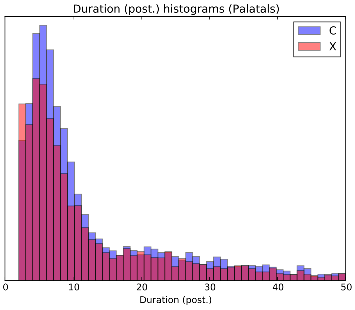




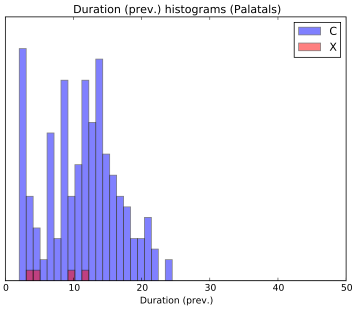
  - Middle phone:

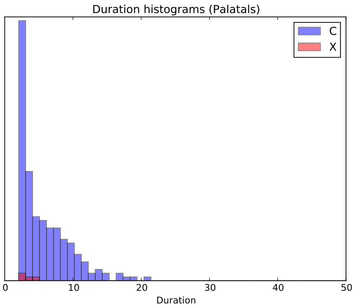




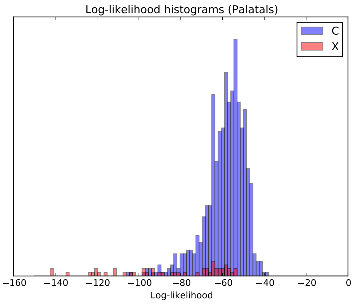


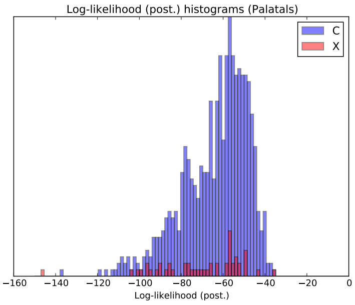
  - Right phone



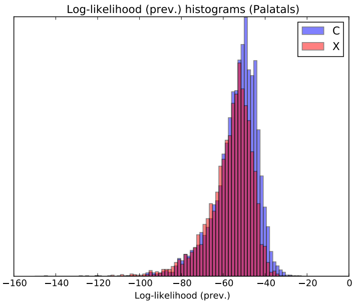


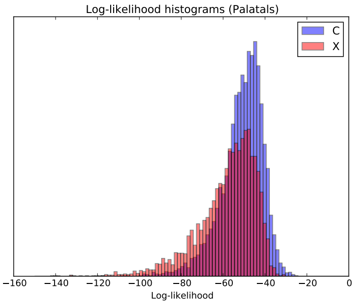
- Log-likelihood:
  - Left phone

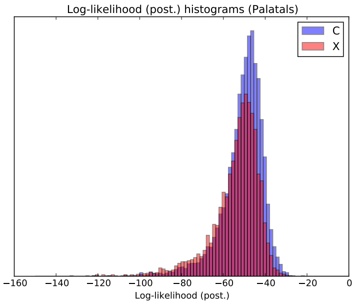




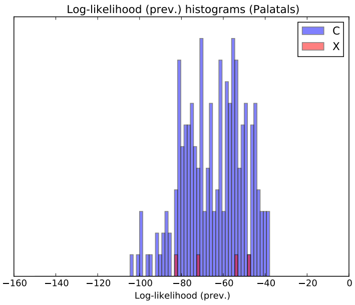
  - Middle phone:

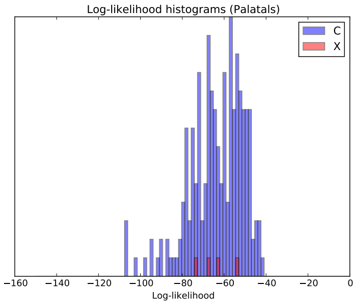






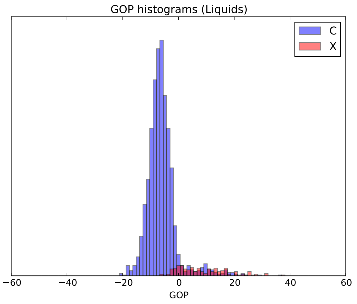
  - Right phone



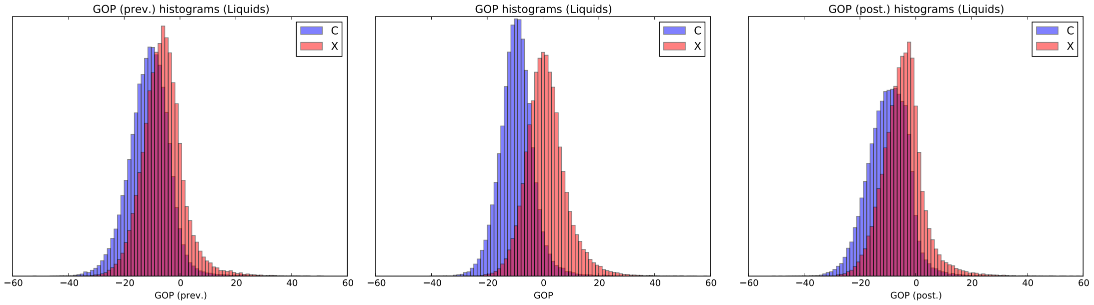


8. **LIQUIDS**

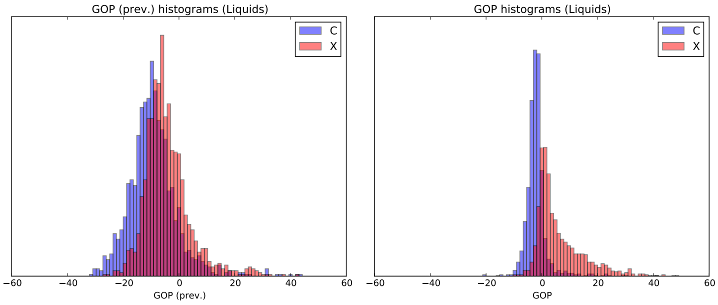
- GOP:
  - Left phone



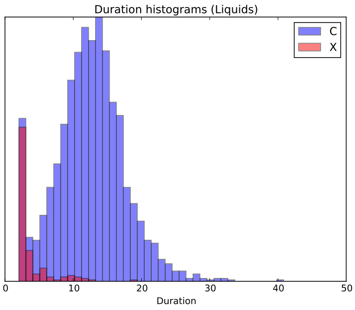
  - Middle phone:

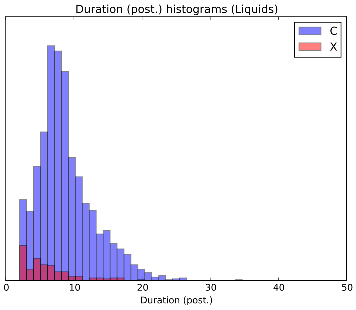


  - Right phone

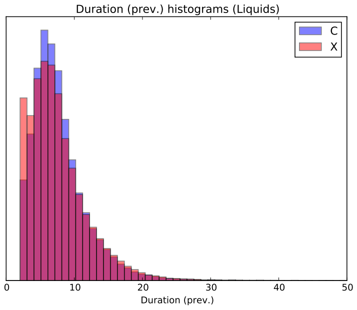


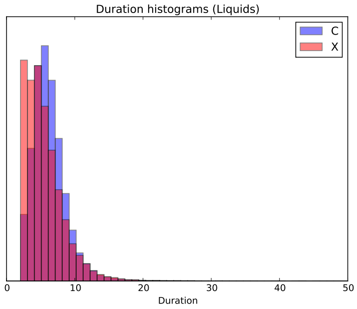
- Duration:
  - Left phone

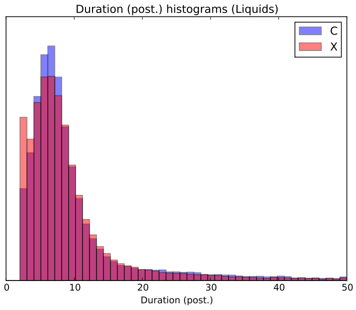




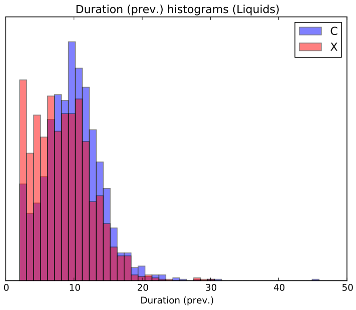
  - Middle phone:

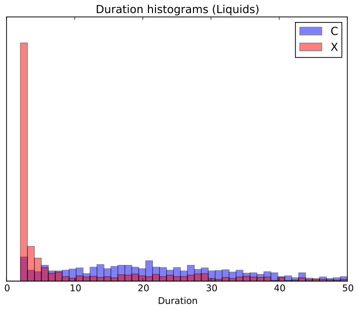




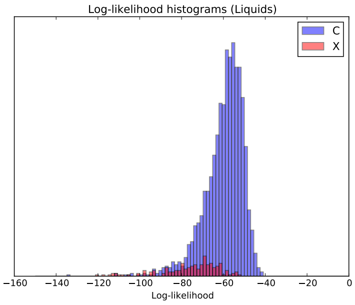


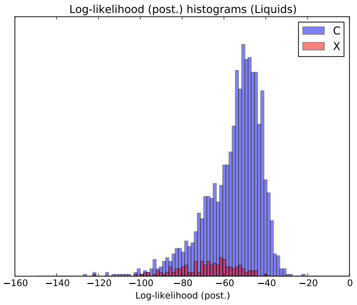
  - Right phone



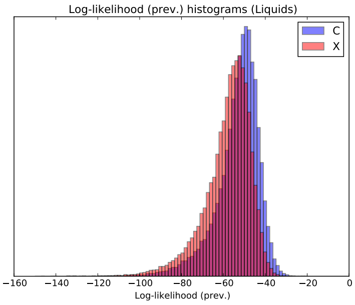


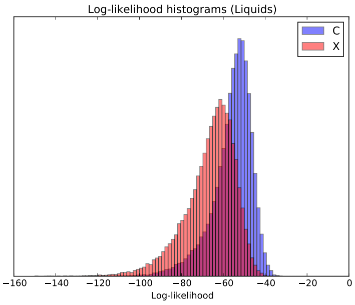
- Log-likelihood:
  - Left phone

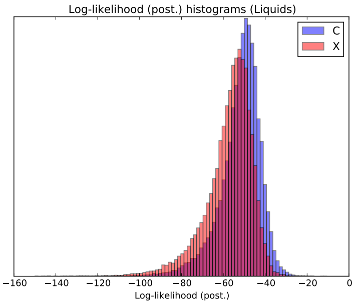




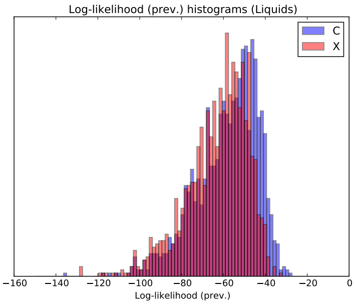
  - Middle phone:

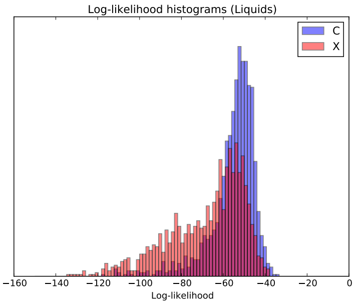






  - Right phone





Bibliography

[1] P. Robinson, *The Routledge Encyclopedia of Second Language Acquisition.* Routledge, 2012.

[2] M. Levy and G. Stockwell, *CALL dimensions: options and issues in Computer-Assisted Language Learning.* Taylor & Francis, 2nd ed., 2013.

[3] B. G. Vice Ministry for Language Policy, "Fifth sociolinguistic survey," 2013.

[4] J. I. Hualde and K. Zuazo, "The standardization of the basque language," *Language Problems and Language Planning*, vol. 31, no. 2, pp. 143–168, 2007.

[5] A. Alberdi and E. A. Batzordea, *Ahoskera.* Hizkuntza Prestakuntza, Eusko Jaurlaritzaren Argitalpen Zerbitzu Nagusia, 2014.

[6] J. Hualde, *Basque Phonology.* Taylor & Francis, 2004.

[7] M. E. Butler-Pascoe, "The history of call: The intertwining paths of technology and secondforeign language teaching," *International Journal of Computer-Assisted Language Learning and Teaching*, vol. 1, no. 1, pp. 16–32, 2011.

[8] J. Harmer, "The practice of english language teaching," *London/New York*, 1991.

[9] C. A. Chapelle, "English language learning and technology," *Language Learning & Language Teaching*, vol. 7, 2003.

[10] C. Lai, "Modeling teachers' influence on learners' self-directed use of technology for language learning outside the classroom," *Computers & Education*, vol. 82, pp. 74–83, 2015.

[11] P. Benson, *Teaching and researching: Autonomy in language learning.* Routledge, 2013.

[12] C. Dorothy, K. Richard, and S. Bryan, "Technology in language use, language teaching, and language learning," *The Modern Language Journal*, vol. 100, no. S1, pp. 64–80, 2016.

[13] R. Kern, P. Ware, and M. Warschauer, *Encyclopedia of Language and Education*, ch. Network-Based Language Teaching, pp. 1374–1385. Springer US, 2008.

[14] S. Herring, D. Stein, and T. Virtanen, *Pragmatics of Computer-Mediated Communication.* Handbooks of Pragmatics (HOPS), De Gruyter, 2013.

[15] S. Fotos and C. Browne, *New Perspectives on CALL for Second Language Classrooms*, ch. Teaching WELL and Loving It. Taylor & Francis, 2013.

[16] O. Viberg and k. Grönlund, "Mobile assisted language learning: A literature review," in *mLearn*, vol. 955, pp. 9–16, 2012.

[17] M. Thomas, H. Reinders, and M. Warschauer, *Contemporary Computer-Assisted Language Learning*, ch. Intelligent CALL. Bloomsbury linguistics, Bloomsbury Academic, 2012.

[18] A. Davies, "Computer-assisted language testing," *CALICO Journal*, vol. 1, no. 5, pp. 41–43, 2013.

[19] D. R. Garrison, *E-learning in the 21st century: A framework for research and practice.* Taylor & Francis, 2011.

[20] N. Nagata, "Computer vs. workbook instruction in second language acquisition," *CALICO Journal*, vol. 14, no. 1, 1996.

[21] M. Eskenazi, "An overview of spoken language technology for education," *Speech Communication*, vol. 51, no. 10, pp. 832–844, 2009.

[22] J. S. Payne and P. Whitney, "Developing l2 oral proficiency through synchronous cmc: Output, working memory, and interlanguage development," *CALICO journal*, vol. 20, pp. 7–32, 01 2002.

[23] U. Felix, "The unreasonable effectivness of call: What have we learned in two decades of research?," *ReCALL*, vol. 20, no. 2, p. 141–161, 2008.

[24] R. Clifford and N. Granoien, *The path of speech technologies in Computer Assisted Language Learning: From research toward practice*, ch. Applications of technology to language acquisition processes: What can work and why, pp. 25–43. Routledge, 2008.

[25] E. M. Golonka, A. R. Bowles, V. M. Frank, D. L. Richardson, and S. Freynik, "Technologies for foreign language learning: a review of technology types and their effectiveness," *Computer Assisted Language Learning*, vol. 27, no. 1, pp. 70–105, 2014.

[26] S. Bodnar, C. Cucchiarini, and H. Strik, “Computer-assisted grammar practice for oral communication,” in *Proc. of International Conference on Computer Supported Education (CSEDU)*, vol. 1, (Noordwijkerhout, Netherlands), pp. 355–361, 2011.

[27] M. Eskénazi, “Using automatic speech processing for foreign language pronunciation tutoring: some issues and a prototype,” *Language Learning & Technology*, vol. 2, no. 2, pp. 67–76, 1999.

[28] W. L. Johnson, S. Marsella, N. Mote, and H. Viljhálmsson, “Tactical language training system: Supporting the rapid acquisition of foreign language and cultural skills,” in *Proc. of InSTILL/ICALL Symposium: NLP and speech technologies in advanced language learning systems (InSTIL/ICALL)*, (Venice, Italy), 2004.

[29] O. Jokisch, U. Koloska, D. Hirschfeld, and R. Hoffmann, “Pronunciation learning and foreign accent reduction by an audiovisual feedback system,” in *Affective Computing and Intelligent Interaction (ACII)*, (Berlin, Germany), pp. 419–425, Springer, Berlin, Heidelberg, 2005.

[30] H. Morton and M. A. Jack, “Scenario-based spoken interaction with virtual agents,” *Computer Assisted Language Learning*, vol. 18, no. 3, pp. 171–191, 2005.

[31] H. Morton and M. Jack, “Speech interactive computer-assisted language learning: a cross-cultural evaluation,” *Computer Assisted Language Learning*, vol. 23, no. 4, pp. 295–319, 2010.

[32] M. Eskenazi, A. Kennedy, C. Ketchum, R. Olszewski, and G. Pelton, “The nativeaccenttm pronunciation tutor: measuring success in the real world,” in *Proc. of the ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, (Farmington, USA), pp. 124–127, International Speech Communication Association (ISCA), 2007.

[33] S. Chevalier, “Speech interaction with saybot, a call software to help chinese learners of english,” in *Proc. of the ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, (Farmington, USA), pp. 37–40, International Speech Communication Association (ISCA), 2007.

[34] D. Grazyna, A. Wagner, N. Cylwik, and O. Jokisch, “An audiovisual feedback system for acquiring l2 pronunciation and l2 prosody,” in *Proc. of the ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, (Warwickshire, UK), International Speech Communication Association (ISCA), 2009.

[35] H. Franco, H. Bratt, R. Rossier, V. Rao Gadde, E. Shriberg, V. Abrash, and K. Precoda, “Eduspeak®: A speech recognition and pronunciation scoring toolkit for computer-aided language learning applications,” *Language Testing*, vol. 27, no. 3, pp. 401–418, 2010.

[36] S. M. Witt, “Automatic error detection in pronunciation training: Where we are and where we need to go,” in *Proc. of International Symposium on Automatic Detection of Errors in Pronunciation Training (IS ADEPT)*, (Stockholm, Sweden), pp. 1–8, International Speech Communication Association (ISCA), 2012.

[37] H. Hamada, S. Miki, and R. Nakatsu, “Automatic evaluation of english pronunciation based on speech recognition techniques,” *IEICE Transactions on Information and Systems*, vol. E76-D, no. 3, pp. 352–359, 1993.

[38] S. M. Hiller, E. Rooney, J. Laver, and M. A. Jack, “Spell: An automated system for computer-aided pronunciation teaching,” *Speech Communication*, vol. 13, no. 3–4, pp. 463–473, 1993.

[39] S. M. A. Goddijn and G. de Krom, “Evaluation of second language learners’ pronunciation using hidden Markov models,” in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Rhodes, Greece), pp. 2331–2334, 1997.

[40] J. Bernstein, M. Cohen, H. Murveit, D. Rtischev, and M. Weintraub, “Automatic evaluation and training in English pronunication,” in *Proc. of International Conference on Spoken Language Processing (ICSLP)*, (Kobe, Japan), pp. 1185–1188, 1990.

[41] L. Neumeyer, H. Franco, M. Weintraub, and P. Price, “Pronunciation scoring of foreign language student speech,” in *Proc. of International Conference on Spoken Language Processing (ICSLP)*, (Philadelphia, USA), pp. 1457–1460, 1996.

[42] G. Kawai and K. Hirose, “A CALL system using speech recognition to train the pronunciation of japanese long vowels, the mora nasal and mora obstruent,” in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Rhodes, Greece), pp. 657–660, 1997.

[43] O. Ronen, L. Neumeyer, and H. Franco, “Automatic detection of mispronunciation for language instruction,” in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Rhodes, Greece), pp. 649–652, International Speech Communication Association (ISCA), 1997.

[44] Y. Kim, H. Franco, and L. Neumeyer, “Automatic pronunciation scoring of specific phone segments for language instruction,” in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Rhodes, Greece), pp. 645–648, International Speech Communication Association (ISCA), 1997.

[45] H. Franco, L. Neumeyer, M. Ramos, and H. Bratt, “Automatic detection of phone-level mispronunciation for language learning,” in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Budapest, Hungary), pp. 851–854, International Speech Communication Association (ISCA), 1999.

[46] S. M. Witt, *Use of speech recognition in Computer-assisted Language Learning.* PhD dissertation, University of Cambridge, Department of Engineering, 1999.

[47] S. M. Witt and S. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Communication*, vol. 30, no. 2-3, pp. 95–108, 2000.

[48] F. Zhang, C. Huang, F. K. Soong, M. Chu, and R. Wang, "Automatic mispronunciation detection for mandarin," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Las Vegas, USA), pp. 5077–5080, Institute of Electrical and Electronics Engineers (IEEE), 2008.

[49] Y. Wang and L. lee, "Improved approaches of modeling and detecting error patterns with empirical analysis for computer-aided pronunciation training," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Kyoto, Japan), pp. 5049–5052, Institute of Electrical and Electronics Engineers (IEEE), 2012.

[50] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 11, (Tokyo, Japan), pp. 49–52, Institute of Electrical and Electronics Engineers (IEEE), 1986.

[51] B.-H. Juang, W. Hou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 3, pp. 257–265, 1997.

[52] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, (Orlando, USA), pp. I–105–I–108, Institute of Electrical and Electronics Engineers (IEEE), 2002.

[53] K. Yan and S. Gong, "Pronunciation proficiency evaluation based on discriminatively refined acoustic models," *International Journal of Information Technology and Computer Science*, vol. 3, no. 2, pp. 17–23, 2011.

[54] X. Qian, F. Soong, and H. Meng, "Discriminative acoustic model for improving mispronunciation detection and diagnosis in computer-aided pronunciation training (capt)," in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Makuhari, Japan), pp. 757–760, International Speech Communication Association (ISCA), 2010.

[55] A. Ito, Y.-L. Lim, M. Suzuki, and S. Makino, "Pronunciation error detection method based on error rule clustering using a decision tree," in *Proc. of European*

Conference on Speech Communication and Technology (EUROSPEECH), (Lisbon, Portugal), pp. 173–176, International Speech Communication Association (ISCA), 2005.

[56] K. Truong, *Automatic Pronunciation Error Detection in Dutch as a Second Language: An Acoustic–phonetic Approach.* PhD dissertation, Utrecht University, Faculty of Humanities, 2004.

[57] H. Strik, K. Truong, F. de Wet, and C. Cucchiarini, "Comparing different approaches for automatic pronunciation error detection," *Speech Communication*, vol. 51, no. 10, pp. 845–852, 2009.

[58] H. Jiang, "Confidence measures for speech recognition: A surve," *Speech Communication*, vol. 45, no. 4, pp. 455–470, 2005.

[59] S. Wei, G. Hu, Y. Hu, and R.-H. Wang, "A new method for mispronunciation detection using support vector machine based on pronunciation space models," *Speech Communication*, vol. 51, no. 10, pp. 896–905, 2009.

[60] J. Jiang and B. Xu, "Exploring the automatic mispronunciation detection of confusable phones for mandarin," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Taipei, Taiwan), pp. 4833–4836, Institute of Electrical and Electronics Engineers (IEEE), 2009.

[61] S. Xu, J. Jiang, Z. Chen, and B. Xu, "Automatic pronunciation error detection based on linguistic knowledge and pronunciation space," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Taipei, Taiwan), pp. 4841–4844, Institute of Electrical and Electronics Engineers (IEEE), 2009.

[62] S.-Y. Yoon, M. Hasegawa-Johnson, and R. Sproat, "Landmark-based automated pronunciation error detection," in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Makuhari, Japan), pp. 614–617, International Speech Communication Association (ISCA), 2010.

[63] K. Hirabayashi and S. Nakagawa, "Automatic evaluation of english pronunciation by japanese speakers using various acoustic features and pattern recognition techniques," in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Makuhari, Japan), pp. 598–601, International Speech Communication Association (ISCA), 2010.

[64] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[65] W. Hu, Y. Qian, and F. Soong, “A new dnn-based high quality pronunciation evaluation for computer-aided language learning (call),” in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Lyon, France), pp. 1886–1890, International Speech Communication Association (ISCA), 2013.

[66] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, “Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers,” *Speech Communication*, vol. 67, pp. 154–166, 2015.

[67] X. Qian, H. Meng, and F. K. Soong, “The use of dbn-hmms for mispronunciation detection and diagnosis in l2 english to support computer-aided pronunciation training,” in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Portland, USA), pp. 775–778, International Speech Communication Association (ISCA), 2012.

[68] I. Odriozola, O. Jokisch, I. Hernaez, and R. Hoffmann, “A pronunciation tutoring system for basque - first development steps,” in *Proc. of ESSV (Elektronische Sprachsignalverarbeitung)*, (Cottbus, Germany), 2012.

[69] J. Bernstein, J. Cheng, and M. Suzuki, “Fluency changes with general progress in l2 proficiency,” in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Makuhari, Japan), International Speech Communication Association (ISCA), 2010.

[70] C. Cucchiarini, H. Strik, and L. Boves, “Quantitative assessment of second language learners’ fluency: an automatic approach,” *The Journal of the Acoustical Society of America*, vol. 107, pp. 989–999, 1998.

[71] F. Hönig, A. Batliner, K. Weilhammer, and E. Nöth, “Islands of failure: employing word accent information for pronunciation quality assessment of english l2 learners,” in *Proc. of the ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, (Warwickshire, UK), International Speech Communication Association (ISCA), 2009.

[72] F. Hönig, A. Batliner, and E. Nöth, “Automatic assessment of non-native prosody – annotation, modelling and evaluation,” in *Proc. of International Symposium on Automatic Detection of Errors in Pronunciation Training (IS ADEPT)*, (Stockholm, Sweden), pp. 21–30, International Speech Communication Association (ISCA), 2012.

[73] A. Bonneau and V. Colotte, “Automatic feedback for l2 prosody learning,” in *Speech and Language Technologies* (I. Ipsic, ed.), pp. 55–70, Intech, 2011.

[74] G.-A. Levow, “Investigating pitch accent recognition in non-native speech,” in *Proc. of the ACL-IJCNLP 2009 Conference Short Papers*, (Stroudsburg, USA), pp. 269–272, Association for Computational Linguistics, 2009.

[75] A. Ito, Y.-L. Lim, M. Suzuki, and S. Makino, “Pronunciation error detection for computer-assisted language learning system based on error rule clustering using a decision tree,” *Acoustical Science and Technology*, vol. 28, no. 2, pp. 131–133, 2007.

[76] H. Ye and S. Young, “Improving speech recognition performance of beginners in spoken conversational interaction for language learning,” in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Lisbon, Portugal), International Speech Communication Association (ISCA), 2005.

[77] O. Saz, E. Lleida, and W. R. Rodríguez, “Acoustic-phonetic decoding for assessment of mispronunciations in speakers with cognitive disorders,” in *Proc. of Advanced Voice Function Assessment (AVFA) International Workshop*, (Madrid, Spain), 2009.

[78] O. Husby, s. Øvregaard, P. Wik, y. Bech, E. Albertsen, S. Nefzaoui, E. Skarpnes, and J. Koreman, “Dealing with l1 background and l2 dialects in norwegian capt,” in *Proc. of the ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, (Venice, Italy), International Speech Communication Association (ISCA), 2011.

[79] A. Neri, C. Cucchiarini, and H. Strik, “Segmental errors in dutch as a second language: how to establish priorities for capt,” in *Proc. of InSTILL/ICALL Symposium: NLP and speech technologies in advanced language learning systems (InSTIL/ICALL)*, (Venice, Italy), 2004.

[80] W. K. Lo, S. Zhang, and H. M. Meng, “Automatic derivation of phonological rules for mispronunciation detection in a computer-assisted pronunciation training system,” in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Makuhari, Japan), International Speech Communication Association (ISCA), 2010.

[81] A. M. Harrison, W. Y. Lau, H. M. Meng, and L. Wang, “Improving mispronunciation detection and diagnosis of learners’ speech with context-sensitive phonological rules based on language transfer,” in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Brisbane, Australia), pp. 2787–2790, International Speech Communication Association (ISCA), 2008.

[82] A. M. Harrison, W.-k. Lo, X.-j. Qian, and H. Meng, “Implementation of an extended recognition network for mispronunciation detection and diagnosis in

computer-assisted pronunciation training," in *Proc. of the ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, (Warwickshire, UK), International Speech Communication Association (ISCA), 2009.

[83] X. Qian, H. Meng, and F. Soong, "On mispronunciation lexicon generation using joint-sequence multigrams in computer-aided pronunciation training (capt)," in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Makuhari, Japan), International Speech Communication Association (ISCA), 2010.

[84] J. M. Norris and L. Ortega, "Effectiveness of l2 instruction: A research synthesis and quantitative meta-analysis," *Language learning*, vol. 50, no. 3, pp. 417–528, 2000.

[85] N. C. Ellis and P. S. Bogart, "Speech and language technology in education: the perspective from sla research and practice.," in *Proc. of the ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, (Farmington, USA), pp. 1–8, International Speech Communication Association (ISCA), 2007.

[86] F. Ehsani, J. Bernstein, A. Najmi, and O. Todic, "Subarashii: Japanese interactive spoken language education," in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Rhodes, Greece), pp. 681–684, International Speech Communication Association (ISCA), 1997.

[87] J. Bernstein, A. Najmi, and F. Ehsani, "Subarashii: Encounters in japanese spoken language education," *CALICO journal*, vol. 16, no. 3, pp. 361–384, 1999.

[88] A. Raux and M. Eskenazi, "Using task-oriented spoken dialogue systems for language learning: potential, practical applications and challenges," in *Proc. of InSTILL/ICALL Symposium: NLP and speech technologies in advanced language learning systems (InSTIL/ICALL)*, (Venice, Italy), 2004.

[89] J. Lee and S. Seneff, "Automatic grammar correction for second-language learners," in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Pittsburgh, USA), pp. 1978–1981, International Speech Communication Association (ISCA), 2006.

[90] H. Wang, C. J. Waple, and T. Kawahara, "Computer assisted language learning system based on dynamic question generation and error prediction for automatic speech recognition," *Speech Communication*, vol. 51, no. 10, pp. 995–1005, 2009.

[91] H. Strik, J. van de Loo, J. van Doremalen, and C. Cucchiarini, "Practicing syntax in spoken interaction: Automatic detection of syntactical errors in non-native utterances," in *Proc. of Interspeech Second Language Studies Workshop*, (Tokyo, Japan), 2010.

[92] H. Strik, J. Colpaert, J. van Doremalen, and C. Cucchiarini, "The disco asr-based call system: practicing l2 oral skills and beyond," in *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, pp. 2702–2707, European Language Resource Association (ELRA), 2012.

[93] B. Penning de Vries, C. Cucchiarini, S. Bodnar, H. Strik, and R. van Hout, "Spoken grammar practice and feedback in an asr-based call system," *Computer Assisted Language Learning*, vol. 28, no. 6, pp. 550–576, 2014.

[94] J. Van Doremalen, H. Strik, and C. Cucchiarini, "Utterance verification in language learning applications," in *Proc. of the ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, (Warwickshire, UK), pp. 13–16, International Speech Communication Association (ISCA), 2009.

[95] G. Bouwman and L. Boves, "Utterance verification based on the likelihood distance to alternative paths," in *Text, Speech and Dialogue*, (Berlin, Heidelberg), pp. 213–220, Springer Berlin Heidelberg, 2002.

[96] I. Odriozola, I. Hernaez, and E. Navas, "Design of a message verification tool to be implemented in call systems," in *Proc. of Advances in Speech and Language Technologies for Iberian Languages (IBERSPEECH)*, (Madrid, Spain), pp. 251–259, 2012.

[97] I. Odriozola, I. Hernaez, and E. Navas, "An on-line VAD based on Multi-Normalisation Scoring (MNS) of observation likelihoods," *Expert Systems with Applications (ESwA)*, vol. 110, pp. 52–61, 2018.

[98] I. Hernaez, I. Luengo, E. Navas, M. Zubizarreta, I. Gaminde, and J. Sanchez, "The basque speechdat (ii) database: a description and first test recognition results," in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Geneva, Switzerland), pp. 1549–1552, International Speech Communication Association (ISCA), 2003.

[99] H. Hoge, H. Tropf, R. Winski, H. van den Heuvel, R. Haeb-Umbach, and K. Choukri, "European speech databases for telephone applications," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Munich, Germany), pp. 1771–1774, Institute of Electrical and Electronics Engineers (IEEE), 1997.

[100] D. Iskra, B. Grosskopf, K. Marasek, H. van den Heuvel, F. Diehl, and A. Kiessling, "Speecon – speech databases for consumer devices: Database specification and validation," in *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, (Las Palmas, Spain), pp. 329–333, European Language Resource Association (ELRA), 2002.

[101] I. Odriozola, I. Hernaez, M. I. Torres, L. J. Rodriguez-Fuentes, M. Penagarikano, and E. Navas, "Basque speecon-like and basque speechdat MDB-600: speech databases for the development of ASR technology for basque," in *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, (Reykjavik, Iceland), pp. 2658–2665, European Language Resource Association (ELRA), 2014.

[102] X. Zalbide, I. Gaminde, I. Hernaez, M. Zubizarreta, and E. Navas, "Euskararako sampa kodeaz," *Euskalingua*, vol. 2, pp. 171–177, 2003.

[103] S. Young, N. Russell, and J. Thornton, "Token Passing: a simple conceptual model for connected speech recognition systems," tech. rep., University of Cambridge, Department of Engineering, 1989.

[104] H. Ney, *Speech recognition and coding, new advances and trends*, ch. Search strategies for Large-Vocabulary Continuous-Speech Recognition, pp. 210–225. NATO ASI Series, 1995.

[105] I. Odriozola, L. Serrano, I. Hernaez, and E. Navas, "The AhoSR automatic speech recognition system," in *Proc. of Advances in Speech and Language Technologies for Iberian Languages (IBERSPEECH)*, (Gran Canaria, Spain), pp. 279–288, 2014.

[106] S. J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*. Cambridge University Press, 2006.

[107] A. Lee and T. Kawahara, "Recent development of open-source speech recognition engine Julius," in *Proc. of Asia-Pacific Signal and Information Processing Association - Annual Summit and Conference (APSIPA ASC)*, (Sapporo, Japan), pp. 131–137, 2009.

[108] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, (Waikoloa, USA), pp. 1–4, Institute of Electrical and Electronics Engineers (IEEE), 2011.

[109] D. Rybach, C. Gollan, G. Heigold, B. Hoffmeister, J. Lööf, R. Schlüter, and H. Ney, "The RWTH aachen university open source speech recognition system," in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Brighton, United Kingdom), pp. 2111–2114, International Speech Communication Association (ISCA), 2009.

[110] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel, "Sphinx-4: A flexible open source framework for speech recognition," tech. rep., Sun Microsystems, 2004.

[111] T. Hirsimäki, M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkönen, "Unlimited vocabulary speech recognition with morph language models applied to finnish," *Computer Speech & Language*, vol. 20, no. 4, pp. 515–541, 2006.

[112] T. Rotovnik, M. S. Maucec, and Z. Kacic, "Large vocabulary continuous speech recognition of an inflected language using stems and endings," *Speech Communication*, vol. 49, no. 6, pp. 437–452, 2007.

[113] H. Sak, M. Saraclar, and T. Güngör, "Morphology-based and sub-word language modeling for turkish speech recognition.," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Dallas, USA), pp. 5402–5405, Institute of Electrical and Electronics Engineers (IEEE), 2010.

[114] G. F. Choueiter, D. Povery, S. F. Chen, and G. Zweig, "Morpheme-based language modeling for Arabic LVCSR," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Toulouse, France), pp. 1053–1056, Institute of Electrical and Electronics Engineers (IEEE), 2006.

[115] P. Mihajlik, T. Fegyó, Z. Tüske, and P. Ircing, "A morpho-graphemic approach for the recognition of spontaneous speech in agglutinative languages - like Hungarian.," in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Antwerp, Belgium), pp. 1497–1500, International Speech Communication Association (ISCA), 2007.

[116] R. Thangarajan, *Modern speech recognition approaches with case studies*, ch. Speech recognition for agglutinative Languages. InTech, 2012.

[117] V. G. Guijarrubia, M. I. Torres, and R. Justo, "Morpheme-based automatic speech recognition of basque," in *Proc. of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, (Póvoa de Varzim, Portugal), pp. 386–393, 2009.

[118] Z. Fang, Z. Guoliang, and S. Zhanjiang, "Comparison of different implementations of mfcc," *Journal of Computer Science and Technology*, vol. 16, no. 6, pp. 582–589, 2001.

[119] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *proc. of IEEE*, (Paris, France), pp. 257–286, Institute of Electrical and Electronics Engineers (IEEE), 1989.

[120] S. J. Young, J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy modelling," in *Proc. of ARPA workshop on Human Language Technology (HLT)*, (Plainsboro, USA), pp. 307–312, 1994.

[121] A. Hunt and S. McGlashan, "Speech recognition grammar specification," tech. rep., World Wide Web Consortium, 2004.

[122] X. Li and Y. Zhao, "A fast and memory-efficient n-gram language model lookup method for large vocabulary continuous speech recognition," *Computer Speech & Language*, vol. 21, no. 1, pp. 1–25, 2007.

[123] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. of International Conference on Spoken Language Processing (ICSLP)*, (Denver, USA), pp. 257–286, 2002.

[124] K. Demuynck, J. Duchateau, D. Van Compernolle, and P. Wambacq, "An efficient search space representation for large vocabulary continuous speech recognition," *Speech Communication*, vol. 30, no. 1, pp. 37–53, 2000.

[125] A. Cardenal, *Realización de un reconocedor de voz en tiempo real para habla continua y grandes vocabularios.* PhD dissertation, University of Vigo, Department of Signal Theory and Communications, 2001.

[126] S. Ortmanns and H. Ney, "Look-ahead techniques for fast beam search," *Computer Speech & Language*, vol. 14, no. 1, pp. 15–32, 2000.

[127] R. Kanters, C. Cucchiarini, and H. Strik, "The Goodness of Pronunciation algorithm: a detailed performance study," in *Proc. of the ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, (Warwickshire, UK), pp. 2–5, International Speech Communication Association (ISCA), 2009.

[128] M. Finke and A. Waibel, "Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition," in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Rhodes, Greece), pp. 2379–2382, International Speech Communication Association (ISCA), 1997.

[129] C. Lopes and F. Perdigao, *Speech technologies*, ch. Phone Recognition on the TIMIT Database, pp. 285–302. Ivo Ipsic, 2011.

[130] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, no. 37, pp. 1641–1648, 1989.

[131] J.-L. Gauvain and L. F. Lamel, "Identification of non-linguistic speech features," in *Proc. of ARPA workshop on Human Language Technology (HLT)*, (Stroudsburg, USA), pp. 96–101, 1993.

[132] T. Robinson, "An application of recurrent nets to phone probability estimation," *IEEE Transactions on Neural Networks*, vol. 5, pp. 298–305, 1994.

[133] L. Tóth, "Modeling long temporal contexts in convolutional neural network-based phone recognition," in *Proc. of IEEE International Conference on Acoustics,*

Speech, and Signal Processing (ICASSP), (South Brisbane, Australia), pp. 4575–4579, Institute of Electrical and Electronics Engineers (IEEE), 2015.

[134] L. Tóth, "Phone recognition with hierarchical convolutional deep maxout networks," *Journal on Audio, Speech and Music Processing*, vol. 2015, p. 25, 2015.

[135] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development.* Upper Saddle River, USA: Prentice Hall PTR, 1st ed., 2001.

[136] I. Luengo, *Análisis y Evaluación de Parámetros para Identificación Automática de Emociones en el Habla.* PhD dissertation, University of the Basque Country, Department of Electronics and Telecommunications, 2010.

[137] I. Odriozola, E. Navas, I. Hernaez, I. Sainz, I. Saratxaga, J. Sánchez, and D. Erro, "Using an ASR database to design a pronunciation evaluation system in basque," in *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, (Istanbul, Turkey), pp. 4122–4126, European Language Resource Association (ELRA), 2012.

[138] I. Odriozola, O. Jokisch, I. Hernáez, and R. Hoffmann, "Diseño y desarrollo de un sistema de evaluación automática de la pronunciación para el euskara," *Procesamiento del Lenguaje Natural*, vol. 49, pp. 101–108, 2012.

[139] B. Mak, M. Siu, M. Ng, Y.-C. Tam, Y.-C. Chan, K.-W. Chan, K.-Y. Leung, S. Ho, F.-H. Chong, J. Wong, and J. Lo, "Plaser: Pronunciation learning via automatic speech recognition," in *Proc. of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing - Volume 2*, (Stroudsburg, USA), pp. 23–29, 2003.

[140] C. for Cultural Co-operation (Education Committee Modern Languages Division), *Common European Framework of Reference for Languages: Learning, Teaching, Assessment.* Applied Linguistics Series, Cambridge University Press, 2001.

[141] P. Adenot, C. Wilson, and C. Rogers, "Web audio api - w3c working draft 10 october 2013," tech. rep., World Wide Web Consortium, 2013.

[142] D. C. Burnett, A. Bergkvist, C. Jennings, and A. Narayanan, "Media capture and streams - w3c last call working draft 14 april 2015," tech. rep., World Wide Web Consortium, 2015.

[143] I. Hickson, "The web sockets api - w3c working draft 22 december 2009," tech. rep., World Wide Web Consortium, 2009.

[144] M. Mustafa, T. Allen, and L. Evett, *Research and Development in Intelligent Systems XXXI*, ch. A Review of Voice Activity Detection Techniques for On-Device

Isolated Digit Recognition on Mobile Devices, pp. 317–329. Springer International Publishing, 2014.

[145] T. Virtanen, R. Singh, and B. Raj, *Techniques for Noise Robustness in Automatic Speech Recognition*. Wiley Publishing, 1st ed., 2012.

[146] I. Mporas, O. Kocsis, T. Ganchev, and N. Fakotakis, "Robust speech interaction in motorcycle environment," *Expert Systems with Applications*, vol. 37, no. 3, pp. 1827–1835, 2010.

[147] E. Principi, S. Squartini, R. Bonfigli, G. Ferroni, and F. Piazza, "An integrated system for voice command recognition and emergency detection based on audio signals," *Expert Systems with Applications*, vol. 42, no. 13, pp. 5668–5683, 2015.

[148] T. Kostoulas, I. Mporas, O. Kocsis, T. Ganchev, N. Katsaounos, J. J. Santamaria, S. Jimenez-Murcia, F. Fernandez-Aranda, and N. Fakotakis, "Affective speech interface in serious games for supporting therapy of mental disorders," *Expert Systems with Applications*, vol. 39, no. 12, pp. 11072–11079, 2012.

[149] S. S. Tirumala, S. R. Shahamiri, A. S. Garhwal, and R. Wang, "Speaker identification features extraction methods: A systematic review," *Expert Systems with Applications*, vol. 90, pp. 250–271, 2017.

[150] T.-W. Kuan, H.-C. Tsai, J.-F. Wang, J.-C. Wang, B.-W. Chen, and Z.-Y. Lin, "A new hybrid and dynamic fusion of multiple experts for intelligent porch system," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9288–9296, 2012.

[151] B. Martínez-González, J. M. Pardo, J. D. Echeverry-Correa, and R. San-Segundo, "Spatial features selection for unsupervised speaker segmentation and clustering," *Expert Systems with Applications*, vol. 73, pp. 27–42, 2017.

[152] J. B. Alonso, J. Cabrera, M. Medina, and C. M. Travieso, "New approach in quantification of emotional intensity from the speech signal: emotional temperature," *Expert Systems with Applications*, vol. 42, no. 24, pp. 9554–9564, 2015.

[153] S. Graf, T. Herbig, M. Buck, and G. Schmidt, "Features for voice activity detection: a comparative analysis," *Journal on Audio, Speech and Music Processing*, vol. 2015, p. 91, 2015.

[154] R. Tucker, "Voice activity detection using a periodicity measure," *IEE Proceedings, Part I: Communications, Speech and Vision*, vol. 4, pp. 377–380, 1992.

[155] V. Hautamäki, M. Tuononen, T. Niemi-Laitinen, and P. Fränti, "Improving speaker verification by periodicity based voice activity detection," in *Proc. of the International Conference on Speech and Computer (SPECOM)*, vol. 2, (Moscow, Russia), pp. 645–650, 2007.

[156] A. Benyassine, “Itu-t recommendation g.729 annex b: a silence compression scheme for use with g.729 optimized for v.70 digital simultaneous voice and data applications,” *IEEE Communications Magazine*, vol. 35, no. 9, pp. 64–73, 1997.

[157] R. Chengalvarayan, “Robust energy normalization using speech/nonspeech discriminator for german connected digit recognition.,” in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Budapest, Hungary), International Speech Communication Association (ISCA), 1999.

[158] L. R. Rabiner and M. R. Sambur, “An algorithm for determining the endpoints of isolated utterances,” *Bell Systems Technical Journal*, vol. 54, no. 2, pp. 297–315, 1975.

[159] P. K. Ghosh, A. Tsiartas, and S. S. Narayanan, “Robust voice activity detection using long-term signal variability.,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 19, no. 3, pp. 600–613, 2011.

[160] Y. Ma and A. Nishihara, “Efficient voice activity detection algorithm using long-term spectral flatness measure,” *Journal on Audio, Speech and Music Processing*, vol. 2013, no. 1, p. 87, 2013.

[161] K. H. Woo, T. Y. Yang, K. J. Park, and C. Lee, “Robust voice activity detection algorithm for estimating noise spectrum,” *Electronic Letters*, vol. 36, no. 2, 2000.

[162] M. Marzinzik and B. Kollmeier, “Speech pause detection for noise spectrum estimation by tracking power envelope dynamics.,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 2, pp. 109–118, 2002.

[163] P. Pollak and P. Sovka, “Cepstral speech/pause detectors,” in *IEEE Workshop on Nonlinear Signal and Image Processing*, (Halkidiki, Greece), pp. 388–391, 1995.

[164] E. Nemer, R. A. Goubran, and S. Mahmoud, “Robust voice activity detection using higher-order statistics in the lpc residual domain.,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 217–231, 2001.

[165] S. G. Tanyer and H. Özer, “Voice activity detection in nonstationary noise,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 478–482, 2000.

[166] J. Tatarinov and P. Pollák, “Hmm and ehmm based voice activity detectors and design of testing platform for vad classification,” *Digital Technologies*, vol. 1, pp. 1–4, 2008.

[167] B. Kingsbury, G. Saon, L. Mangu, M. Padmanabhan, and R. Sarikaya, “Robust speech recognition in noisy environments: The 2001 ibm spine evaluation system,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Orlando, USA), pp. I–53–I–56, Institute of Electrical and Electronics Engineers (IEEE), 2002.

[168] H. Veisi and H. Sameti, “Hidden-Markov-model-based voice activity detector with high speech detection rate for speech enhancement,” *IET Signal Processing*, vol. 6, no. 1, pp. 54–63, 2012.

[169] s. Varela, R. San-Segundo, and L. A. Hernández, “Combining pulse-based features for rejecting far-field speech in a hmm-based voice activity detector,” *Computers and Electrical Engineering*, vol. 37, no. 4, pp. 589–600, 2011.

[170] D. Enqing, L. Guizhong, Z. Yatong, and C. Yu, “Voice activity detection based on short-time energy and noise spectrum adaptation,” in *Proc. of IEEE International Conference on Signal Processing (ICSP)*, (Beijing, China), p. 464–467, Institute of Electrical and Electronics Engineers (IEEE), 2002.

[171] J. Ramirez, P. Yelamos, J. Gorriz, J. Segura, and L. Garcia, “Speech/non-speech discrimination combining advanced feature extraction and svm learning,” in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Pittsburgh, USA), pp. 1662—1665, International Speech Communication Association (ISCA), 2006.

[172] J. Ramirez, P. Yelamos, J. M. Gorriz, and J. C. Segura, “SVM-based speech endpoint detection using contextual speech features,” *Electronic Letters*, vol. 42, no. 7, pp. 426–428, 2006.

[173] Y. W. Tan, W. J. Liu, W. Jiang, and H. Zheng, “Hybrid svm/hmm architectures for statistical model-based voice activity detection,” in *Proc. of International Joint Conference on Neural Networks (IJCNN)*, (Anchorage, USA), pp. 2875–2878, 2014.

[174] T. Hughes and K. Mierle, “Recurrent neural networks for voice activity detection,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Vancouver, Canada), pp. 7378–7382, Institute of Electrical and Electronics Engineers (IEEE), 2013.

[175] S. Thomas, S. Ganapathy, G. Saon, and H. Soltau, “Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Florence, Italy), pp. 2519–2523, Institute of Electrical and Electronics Engineers (IEEE), 2014.

[176] Y. Obuchi, “Framewise speech-nonspeech classification by neural networks for voice activity detection with statistical noise suppression,” in *ICASSP*, (Shanghai, China), pp. 5715–5719, Institute of Electrical and Electronics Engineers (IEEE), 2016.

[177] A. Sehgal and N. Kehtarnavaz, “A convolutional neural network smartphone app for real-time Voice Activity Detection,” *IEEE Access*, vol. 6, pp. 9017–9026, 2018.

[178] X. Zhang and J. Wu, “Deep belief networks based Voice Activity Detection,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 21, pp. 697–710, 2013.

[179] I. Luengo, E. Navas, I. Odriozola, I. Saratxaga, I. Hernáez, I. Sainz, and D. Erro, “Modified LTSE-VAD algorithm for applications requiring reduced silence frame misclassification.,” in *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, (Valletta, Malta), pp. 1539–1544, European Language Resource Association (ELRA), 2010.

[180] B. Kotnik, P. Sendorek, S. Astrov, T. Koç, T. Çiloglu, L. Docío Fernández, E. Rodríguez Banga, H. Höge, and Z. Kacic, “Evaluation of voice activity and voicing detection,” in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Brisbane, Australia), pp. 1642–1645, International Speech Communication Association (ISCA), 2008.

[181] D. Macho, L. Mauuary, B. Noé, Y. M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, “Evaluation of a noise-robust dsr front-end on aurora databases,” in *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (Denver, USA), International Speech Communication Association (ISCA), 2002.

[182] J. Ramirez, J. C. Segura, C. Benitez, A. De La Torre, and A. Rubio, “Efficient voice activity detection algorithms using long-term speech information,” *Speech Communication*, vol. 42, pp. 3–4, 2004.

[183] M. Westphal, “The use of cepstral means in conversational speech recognition.,” in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Rhodes, Greece), International Speech Communication Association (ISCA), 1997.

[184] S. F. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[185] P. N. Garner, “Cepstral normalisation and the signal to noise ratio spectrum in automatic speech recognition,” *Speech Communication*, vol. 53, no. 8, pp. 991–1001, 2011.

[186] F.-H. Liu, R. M. Stern, X. Huang, and A. Acero, “Efficient cepstral normalization for robust speech recognition,” in *Proc. of ARPA workshop on Human Language Technology (HLT)*, (Stroudsburg, USA), pp. 69–74, 1993.

[187] F.-H. Liu, R. Stern, A. Acero, and P. Moreno, “Efficient cepstral normalization for robust speech recognition,” in *Proc. of IEEE International Conference on*

Acoustics, Speech, and Signal Processing (ICASSP), (Adelaide, Australia), Institute of Electrical and Electronics Engineers (IEEE), 1994.

[188] B. Widrow, R. G. Winter, and R. A. Baxter, "Layered neural nets for pattern recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 110.7, pp. 1109–1118, 1988.

[189] W. H. Delashmit and M. T. Manry, "Recent developments in multilayer perceptron neural networks," in *Proc. of the 7th Annual Memphis Area Engineering and Science Conference (MAESC)*, (Memphis, USA), 2005.

[190] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Volume 1: Foundations* (D. E. Rumelhart, J. L. McClelland, *et al.*, eds.), pp. 318–362, Cambridge: MIT Press, 1987.

[191] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Washington: Spartan Books, 1962.

[192] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry.* Cambridge, USA: MIT Press, 1969.

[193] R. Collobert and S. Bengio, "Links between perceptrons, MLPs and SVMs," in *Proc. of International Conference on Machine Learning (ICML)*, (Banff, Canada), 2004.

[194] G. Holmes, A. Donkin, and I. H. Witten, "Weka: a machine learning workbench," in *Proc. of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems*, pp. 357–361, August 1994.

[195] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[196] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, and D. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon Technical Report N*, vol. 93, Feb. 1993.

[197] A. Abdulaziz and V. Kepuska, "Noisy timit speech (ldc2017s04)," 3 2017.

[198] I. T. Union, "Generic sound activity detector (gsad); series g: Transmission systems and media, digital systems and networks: Digital terminal equipments-coding of voice and audio signals. g.720.1," tech. rep., Telecommunication standardization sector of ITU (ITU-T), 2010.

[199] I. T. Union, "Coding of speech at 8 kbit/s using conjugate structure algebraic-code-excited linear prediction (cs-acelp); series g: Transmission systems and media, digital systems and networks: Digital terminal equipments-coding of voice and audio signals. g.729," tech. rep., Telecommunication standardization sector of ITU (ITU-T), 2012.

[200] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[201] O. Viikki and K. Laurila, "Noise robust hmm-based speech recognition using segmental cepstral feature vector normalization," in *Robust Speech Recognition for Unknown Communication Channels*, (Pont-à-Mousson, France), pp. 107–110, ISCA, 1997.

[202] S. Tibrewala and H. Hermansky, "Multi-band and adaptation approaches to robust speech recognition," in *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)*, (Rhodes, Greece), International Speech Communication Association (ISCA), 1997.

[203] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, no. 1-3, pp. 133–147, 1998.

[204] O. M. Strand and A. Egeberg, "Cepstral mean and variance normalization in the model domain," 2004.

[205] N. V. Prasad and S. Umesh, "Improved cepstral mean and variance normalization using bayesian framework.," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, (Olomouc, Czech Republic), pp. 156–161, Institute of Electrical and Electronics Engineers (IEEE), 2013.

[206] D. Willett, "Online maximum-likelihood mean and variance normalization for speech recognition," August 2015. US Patent App. 14/640,912.

[207] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[208] C.-P. C. Karim, C.-p. Chen, K. Filali, and J. A. Bilmes, "Frontend post-processing and backend model enhancement on the aurora 2.0/3.0 databases," in *Proc. of International Conference on Spoken Language Processing (ICSLP)*, (Denver, USA), pp. 241–244, 2002.

[209] P. Pujol, D. Macho, and C. Nadeu, "On real-time mean-and-variance normalization of speech recognition features," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Toulouse, France), Institute of Electrical and Electronics Engineers (IEEE), 2006.

[210] O. Viikki, D. Bye, and K. Laurila, "A recursive feature vector normalization approach for robust speech recognition in noise.," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Seattle, USA), pp. 733–736, Institute of Electrical and Electronics Engineers (IEEE), 1998.

[211] M. Ashby and J. Maidment, *Introducing Phonetic Science.* Cambridge Introductions to Language and Linguistics, Cambridge University Press, 2014.

[212] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.