Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

A Pronunciation Scoring System for Second Language Learners

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Federico Nicolás Landini

Directora: Dra. Luciana Ferrer

Buenos Aires, Febrero 2017

ABSTRACT

In computer-assisted language learning (CALL) systems, one of the tasks involved is pronunciation scoring which aims to automatically detect pronunciation problems, allowing the system to provide valuable feedback to the student. Pronunciation scoring systems give, for an utterance or set of utterances, a score saying how correct or native was the pronunciation.

In this work we present an automatic pronunciation scoring system based on two different methods that we will call LLK and LLR, since they are based on log-likelihoods and log-likelihood ratios of automatic speech recognition (ASR) models trained with different populations of speakers. The LLK score is computed as the log-likelihood of an utterance with respect to a model trained only with native speakers. A badly pronounced utterance is then expected to generate low scores, while a native-like utterance should fit the model well and, hence, generate higher scores. The LLR score is computed as the subtraction of the log-likelihoods of an utterance with respect to a model trained with non-native speakers and a model trained with native speakers. This score gives information about what model is "closer" to the utterance pronunciation. A small value for the score corresponds to a "more native" pronunciation, while a large value corresponds to a "more non-native" pronunciation. The proposed LLR method is based on an approach previously introduced for the related task of mispronunciation detection at phone level. In this work, we adapt this successful approach to the task of pronunciation scoring, which required some significant changes in the method.

In this work we assume that not enough data matched to the test conditions is available to train the native and non-native ASR models from scratch. Hence, an initial model is trained on a large dataset of native speakers consisting of conditions mismatched to those in the test set. This model is then adapted to two population of speakers, natives and non-natives, collected under the same conditions as the test data.

The LLK method gave worse results than expected in comparison with published results. We believe this is because the corpora were composed of telephone conversations with high levels of noise and distortion, a condition rarely considered in the pronunciation scoring literature. On the other hand, our implementation of the LLR method gave competitive results compared to those in the literature. We believe this method does not suffer from noise or distortion in the same degree as LLK because the log-likelihoods obtained with both models are influenced in the same manner by noise and distortion. When subtracting the two log-likelihoods, the effect of the acoustic conditions partly compensate each other, therefore reducing the influence of signal distortions. The evaluation of LLR in a held-out set of data obtained a correlation of 0.77 with human ratings at speaker level (several sentences). These results are comparable with published results when testing on similar data.

Keywords: Computer-Assisted Language Learning, Pronunciation Scoring, Automatic Speech Recognition, Gaussian Mixture Models, Native and Non-Native Speech, Log-Likelihood Ratio, MAP adaptation.

RESUMEN

En los sistemas de asistencia computarizada para aprendizaje de idiomas (ACAI), una de las tareas es puntuar la pronunciación con el fin de detectar problemas de pronunciación de manera automática y dar *feedback* valioso a los estudiantes de idiomas. Los sistemas de puntuación de la pronunciación dan, dado un segmento de habla, un puntaje indicando cuán correcta o nativa fue la pronunciación.

En este trabajo presentamos un sistema automático de puntuación de la pronunciación basado en dos métodos diferentes que llamamos LLK y LLR, dado que están basados en log-likelihoods y en un log-likelihood ratio. Dichos log-likelihoods son obtenidos con sistemas de reconocimiento de habla automáticos (RHA) entrenados con diferentes poblaciones de hablantes. El puntaje LLK se computa como el log-likelihood de un segmento de habla respecto de un modelo entrenado solamente con hablantes nativos. De esta manera, se espera que un segmento de habla mal pronunciado produzca puntajes bajos mientras que se espera que habla nativa se ajuste al modelo y, por ende, produzca puntajes altos. El puntaje LLR se computa como la resta entre los log-likelihoods de un segmento de habla respecto de un modelo entrenado con hablantes nativos y de un modelo entrenado con hablantes no nativos. Este puntaje indica qué modelo es "más cercano" a la pronunciación en el segmento de habla. Un valor bajo se corresponde con una pronunciación "más nativa" mientras que un valor alto se corresponde con una pronunciación "más no nativa". El método LLR propuesto está basado en un enfoque previamente introducido para la tarea relacionada de detección de errores en la pronunciación a nivel fono. En este trabajo, adaptamos este exitoso enfoque a la tarea de puntuación de la pronunciación que requirió cambios significativos en el método.

En este trabajo asumimos que no hay suficientes datos de hablantes nativos y no nativos para evaluar el sistema y entrenar los modelos de reconocimiento de habla desde cero. Por lo tanto, un modelo de reconocimiento de habla es entrenado inicialmente sobre un gran conjunto de datos de hablantes nativos con condiciones acústicas diferentes a las del conjunto de evaluación. Este modelo es luego adaptado a dos poblaciones de hablantes, nativos y no nativos, recolectados bajo las mismas condiciones que el conjunto de datos de evaluación.

El método LLK dio peores resultados que los esperados en comparación con resultados publicados. Creemos que esto se debe a que los corpora estaban compuestos por conversaciones telefónicas con altos niveles de ruido y distorsión, una condición rara vez considerada en la literatura de puntuación de la pronunciación. Por otro lado, nuestra implementación del método LLR dio resultados competitivos en comparación con aquellos vistos en la literatura. Creemos que este método no sufre con el ruido o la distorsión de la misma manera que LLK porque los log-likelihoods obtenidos con ambos modelos son influenciados en la misma manera por ruido y distorsión. Al restar los dos log-likelihoods, el efecto de las condiciones acústicas se compensan en parte, reduciendo la influencia de distorsiones en la señal. La evaluación de LLR en un conjunto de datos de validación final obtuvo una correlación de 0.77 con puntajes manualmente asignados a nivel hablante (varias oraciones). Estos resultados son comparables con aquellos publicados donde se evaluó en datos de similares características.

Palabras claves: Asistencia Computarizada para Aprendizaje de Idiomas, Puntuación de la Pronunciación, Reconocimiento de Habla Automático, Modelos de Mezclas de Gaussianas, Habla Nativa y No Nativa, Log-Likelihood Ratio, Adaptación MAP.

AGRADECIMIENTOS

A mi directora, Luciana Ferrer, por guiarme durante los últimos meses con gran dedicación, paciencia y por hacer que el trabajo al que le tenía mucha expectativa fuera un placer. Por enseñarme, no solamente con las discusiones más técnicas sino también con su ejemplo.

A mis padres, por apoyarme durante la carrera, fomentar desde chico mi curiosidad y mi lado científico y darme la oportunidad de elegir mis estudios sin presiones. Al resto de mi familia, por las reuniones a las que me ausenté por exámenes o trabajos.

A mi país, Argentina, uno de los pocos países en el mundo en ofrecer educación universitaria no arancelada y de alta calidad.

A mis dos compañeros de aventuras durante la carrera, Laouen Belloli y Pablo "Partu" Artuso. Sin dudas no habría disfrutado hacer tantos tps si no fuera por hacerlos juntos y con discusiones de todo tipo. Por haberme permitido aprender *con* y *de* ellos y mostrarme la gran diferencia entre "grupo" y "equipo".

Al cuarto mosquetero, Roberto Rama, con quien compartí(mos) la mayoría de los trabajos del final de la carrera y con quien trabajar también siempre fue un gusto.

A Sacha Kantor y al "Dream Team", Franco Castellacci, Gabriel Defagot, Gonzalo Lera Romero y Victor Wjugow por las cursadas juntos, las discusiones dentro y fuera de la facultad y por los asados forjadores de amistad.

Al equipo de Métodos Numéricos por enseñarme mucho de la docencia y permitirme proponer y llevar a cabo ideas. Sin dudas mi formación estaría mucho más lejos de ser completa sin la docencia y los docentes de Métodos me ayudaron a crecer en ese aspecto.

A mis amigos de la infancia, Ema, Fabri y Juanma y los de la escuela, Fer, Marco y Pablo por entender mis tiempos durante las cursadas pero aún así mantener el contacto durante estos años.

A todos los que me acompañaron y me dieron su apoyo durante estos años.

CONTENTS

1. INTRODUCTION

Second language acquisition is a time-consuming process that requires a certain degree of effort on part of both student and teacher. Several techniques have been studied through the years to make this task easier. One important part of learning a new language consists in achieving good pronunciation. This is the subtask studied in this work.

1.1 Problem Description and Motivation

Speech processing has empowered a myriad of technology developments among which are Computer-Assisted Language Learning (CALL) systems. These systems offer software tools to students to learn and practise a diverse set of abilities. In a globalised world where learning languages is increasingly important and the access to computers is increasingly popular, the use of such systems allows students to practise languages more easily. CALL programs give the opportunity to practise without a human teacher and at the student pace, allowing them to repeat exercises at odd hours or frequencies with a tireless teacher.

The use of speech recognition techniques can be applied to CALL allowing students to record pieces of speech, returning a transcription of what has been said and a score exhibitting pronunciation quality and/or grammar correctness. Currently, there are programs that offer this kind of evaluation but usually without taking into account the native language of the learner. However, it has been shown that different mispronunciation errors are produced by students with different native language (L1) and target language (L2) pairs. Furthermore, these errors are not only a consequence of the phonological differences between L1 and L2 but also of the phoneme-grapheme correspondences in both L1 and L2. For example, English learners whose native language is Spanish may pronounce the *z* in *zero* as a voiceless alveolar fricative (/s/[1]) instead of a voiced one (/z/) due to phonological differences, while English learners whose native language is German may mispronounce the *v* in *very* as a voiceless labio-dental fricative (/f/) instead of a voiced one (/v/) due to phoneme-grapheme correspondence. Taking this into account, programs that offer generic language courses (without knowledge of the student's L1) lack the chance to give a more tailored score and instructions on how to overcome the mistakes.

In this work we consider a scenario where the L1 is known to the system and some amount of transcribed data is available for this population of speakers. The development of a CALL system can cover a wide range of aspects including grammar and listening exercises, among others. The focus here will be on generating a score for a fragment of speech reflecting the quality of pronunciation. This score will be obtained as the log-likelihood ratio of the features within the fragment given two Automatic Speech Recognition (ASR) models, one adapted to native speakers and the other to non-native speakers. This method was selected because it was shown in the bibliography that it produced some of the best results. Although this work focuses on Spanish as L1 and English as L2, the same approach can be extrapolated to different L1-L2 pairs with a similar analysis.

1.2 Previous Work on Pronunciation Asessment

When assessing learners' pronunciation, several aspects can be taken into account. The two most relevant categories are prosodic and spectral. The first one regards rhythm, melody and stress. Examples of these are the intonation when asking questions or the stress in different words

[1] According to the alphabetic system of phonetic notation *International Phonetic Alphabet* (IPA).

used for giving emphasis. These aspects give important clues regarding the level of nativeness of a speaker and play an important role for expert human listeners when judging a non-native speaker's pronunciation [2].

The spectral aspects of pronunciation relate to the articulation of individual phones as well as their combination. Previous work regarding this aspect will be discussed in more detail later in this section.

As with the development of any automatic annotation system, the development of a pronunciation scoring system requires ground-truth annotations for training the system and evaluating its quality. To obtain such annotations, the spoken utterances are jugded by expert human raters, for instance linguists, phoneticians or, less frequently, language teachers (in [6] it is claimed that speech therapists may address pronunciation problems more appropiately than teachers). One of the first works producing this kind of benchmark in pronunciation assessment is [1] which is also one of the first articles that focused on spectral features to grade pronunciation. There is high agreement on this method for creating the ground truth labels since the human judgement is what the software is intended to reproduce. Since the amount of utterances needed to train a recognition system with good performance is considerably large, the creation of a labelled dataset is a time consuming, hence expensive, task. A good review on the annotation process is presented by Hönig et al. in [23] where they also evaluate "turkers" (see [46]) as raters.

For this work a single Spanish native speaker (the author of this thesis) with an advanced level of English as second language rated speech. The work here presented was part of a bigger project[2] for which data was not yet collected. In order to develop the pronunciation scoring methods in the midtime, other data, which required labelling, was used. Because it was not the project data, it was not possible to have it rated by paid labellers. Thus, only one person labelled it for free.

Pronunciation assessment methods can be classified in two groups depending on whether non-native data is used for training the system or not, as explained in [23]. Those based only on native data propose models that describe what is an *acceptable* pronunciation (see [1], [36], [29], [13], [7], [6], [48], [53], [5], [4], [47], [57], [3], [9], [24], [28], [23], [30]). The systems measure how *similar* is an utterance in comparison to a model of native speech. The main advantage of this approach is that only native speech is used. Furthermore, no error annotations are necessary to train the model in this case.

On the other hand, models based on native and non-native data or based on knowledge of common mistakes usually made by second language learners describe both *acceptable* and *unacceptable* pronunciations (see [45], [12], [20], [14], [53], [4], [25], [49], [34], [3], [51], [18], [27], [54], [55], [40], [28], [50], [11], [43]). Although these approaches have shown better results than those based only on native models, data collection is much more expensive because enough annotated non-native speech with both good and bad pronunciation labels is needed.

Many pronunciation assessment systems rely on the use of ASR systems and some of the first successful applications date to the early 1990's ([1], [17], [21], [44]). The task of ASR is to find the sequence of words that maximizes the posterior probability given the features extracted from the speech. The two usual approaches for using ASR for pronunciation assessment utilize the posterior probability or the likelihood of the recognized words. ASR models will be further discussed in 2.1.

One important difference in the models that use ASR to assess pronunciation is related to knowledge of the text. Models based on fixed text prompts are referred as "text dependent" and rely on statistics related to specific words, phrases, or sentences. This approach was able to obtain good correlation with human raters ([1]) but lacks of flexibility since adding new

[2] Proyecto de Investigación Científica y Tecnológica funded by Agencia Nacional de Promoción Científica y Tecnológica of Ministerio de Ciencia Tecnología e Innovación Productiva and leaded by Dra. Ferrer.

lessons needs additional data collection to be incorporated into the model. On the other hand, "text independent" models use a reliable speech recognizer on the student speech to detect what was said and the transcription obtained is fed to the pronunciation scoring system. This approach allows new lessons to be designed without changing the implemented model giving more flexibility to application developers.

In this work, we assume that the speech transcripts are available for scoring. This means that the system is not text independent. However, the system does not need to know the text at training time and have examples for it. The transcripts presented to the system can be, for example, selected by a teacher or can be part of a large set which is not necessarily used to train the system. Therefore, the system here presented is in between the two classes aforementioned, having the flexibility to add new lessons without changing the model and only providing the text that the students should read along with the recording at scoring time.

In this work, a pronunciation scoring system based on ASR models is presented and thus only previous publications using ASR are reviewed. However, there are pronunciation assessment models that do not rely on ASR systems and most of them are based on features extracted directly from the signal. Some of the first experiments providing pronunciation feedback using prosodic features using the fundamental frequency (F0) were carried out in the seventies and eighties. Some more recent examples can be found in [48] to generate prosodic features to assess degree of nativeness, in [47] for classification of velar voiceless fricatives and velar voiceless plosives, in [9] for detection of vowel pronunciation errors, or in [28] for pronunciation score and intelligibility. An overview on different approaches among which are features obtained without an ASR model can be found in [52]. Bibliography on prosodic features will not be further discussed for not being the focus of this work, but more information can be found in [10].

In pronunciation assessment two subtasks can be named, *mispronunciation detection* and *pronunciation scoring*. The first one aims to identify pronunciation problems, rather than return a score for an utterance, in order to give relevant feedback to the language student about the mistake(s) (see [45], [14], [20], [53], [49], [47], [57], [9], [27], [51], [3], [50], [11], [43]) while the second one focuses on returning a score describing how good or bad is a speech segment in terms of pronunciation quality, nativity or intelligibility (see [1], [36], [13], [6], [7], [48], [34], [3], [23]).

Although these two subtasks seem very similar, they differ not only on the kind of feedback they give to students but also on the level evaluated (usually phones in mispronunciation detection and sentences in pronunciation scoring). However, some of the techniques designed for one subtask can be used for the other one as is the case with the log-likelihood ratio used for this work which was first devised for mispronunciation detection but here is used for pronunciation scoring.

Several approaches have been studied during the last 25 years and it is not the purpose of this work to review them all. Some interesting work reviews are presented by Neumeyer et al. [35] explaining different methods proposed in the 1990's and by Witt [52] revising a myriad of approaches as of year 2012. Only the relevant approaches for this work are reviewed in the next subsections.

1.2.1 Mispronunciation Detection Review

As was previously mentioned, the aim of mispronunciation detection is to give feedback on speech pronunciation by expressing if a fragment of speech was correctly pronounced or not. Although this approach could be applied to words, it is usually applied to smaller fragments such as phones. In this subsection we review those publications related to the specific scoring techniques used in this work.

The approach used for this work, which we call log-likelihood ratio (LLR), is based on two models, one trained with native speakers and the other with non-native speakers. This is one of the state-of-the-art techniques in pronunciation assessment.

The idea of using non-native speech as well as native speech for training the system was proposed in [45] by Ronen et al. in a mispronunciation detection task. They used a pronunciation network (capturing the word to phones mapping) with both native and non-native pronunciations. Thus, these mispronunciation networks did not only capture the native lexicon but also non-native mistakes. Their idea was then to compute the mispronounced phones rate (number of non-native phones over all) to identify pronunciation problems.

The second approach studied in that paper was based on two models, both with linear pronunciation networks but one using native pronunciations and the other with non-native ones. Then, the only difference between the two models was the pronunciation network they used. For each utterance two forced alignments were done, one with each model. The likelihood scores from these forced alignments were used to compute a likelihood ratio. This method obtained better results than the mispronunciation network approach and comparable to the state-of-the-art technique at that time which used the posterior probability of utterances after doing forced alignment.

As a continuation to that work, in [14] two mispronunciation detection methods were presented. One of them used the posterior likelihood, while the other used two GMM models for each phone. One GMM model was trained with native phone pronunciations and the other trained with non-native or "mispronounced" pronunciations of the same phone. A length-normalized log-likelihood ratio score was computed for the phone segment q_i based on the "mispronounced" model λ_M and the "correct" model λ_C as follows:

$$LLR(q_i) = \frac{1}{d} \sum_{t=t_0}^{t_0+d-1} [\log p(y_t|q_i, \lambda_M) - \log p(y_t|q_i, \lambda_C)], \tag{1.1}$$

where d is the duration in frames of the phone q_i, y_t is the observation vector at frame t and t_0 is the frame where that phone starts. Normalizing by d allows definition of unique thresholds per phone, independent of its length. When the LLR is above that threshold, the phone is detected as a mispronunciation. The other method computed a score based on the log-posterior probability for each phone segment with label q_i. The posterior probability $P(q_i|y_t)$ of the phone q_i given the observation vector y_t is computed for each frame as

$$P(q_i|y_t) = \frac{p(y_t|q_i)P(q_i)}{\sum\limits_{j=1}^{M} p(y_t|q_j)P(q_j)}, \tag{1.2}$$

where j runs over a set of context-independent models for all phone classes and $P(q_i)$ is the prior probability of the phone class q_i. The posterior-based method for the phone segment is obtained as follows:

$$\rho(q_i) = \frac{1}{d} \sum_{t=t_0}^{t_0+d-1} \log P(q_i|y_t), \tag{1.3}$$

where d is the frame duration of the phone and t_0 is the starting frame index of the phone segment. That score is compared to a threshold to determine if the phone pronunciation was correct or not.

Overall, the LLR method performed better than the posterior-based method when compared to phones' transcriptions from experts and the greater gains were observed with the phone classes with the lowest detection error. This was the first work to use the ratio between a native and a non-native model and showed the potential of that approach.

As a continuation to that work, in [11], a common model was first trained using all speech (data from both native and non-native speakers). Then, to obtain the "correct" and "mispronounced" models, the common model was adapted to native speakers' data to obtain the native model and to non-native speakers' data to obtain a non-native model. GMM means and mixture weights were adapted to the class specific data. Thus, instead of training two separate models for each phone from scratch, they were obtained using a common model adapted to each population. They compared the LLR method as in [45], that used two models trained from scratch, with the LLR with adaptation method. The adaptation method obtained a better performance for almost all phones and the major gains were seen with phone classes with smaller amounts of training data.

Finally, a recent work by Robertson et al. [43] followed the line here described. They focused their study in adult beginner learners and proposed a mispronunciation detection method at word level instead of phone level. The methods revised were six and the best results were for a method based on [11]. These findings show that the adaptation to native and non-native data to compute LLR can be considered state-of-the-art for pronunciation assessment.

Nevertheless, the method proposed in this work presents some differences to this successful methods for mispronunciation detection. Our method is based on the log-likelihoods obtained with two ASR models as the method in [45] instead of GMM models as in [11] but obtains the ASR models by adaptation as in [11].

1.2.2 Pronunciation Scoring Review

Although the works mentioned in the previous subsection were focused on mispronunciation detection, those ideas can be used also for pronunciation scoring. In this subsection we review those works related to the two methods evaluated in this work: log-likelihood (LLK) and LLR. In pronunciation scoring, the performance is measured computing the level of correlation between the system output and the raters. It is common to measure correlation for sentence and speaker level ([36], [13]). The amount of speech at speaker level comprises several sentences and thus they are usually more reliable for having more information about the speaker.

The works by Neumeyer et al. and Franco et al. ([36], [13]) were prior to the work that proposed the log-likelihood ratio approach. Both of them used the same corpus in which all non-native speech was read or imitated and most of native speech was read. Speech was collected in a quiet room and with good quality equipment. Neumeyer et al. analyzed different pronunciation methods among which was the log-likelihood. Although other methods had better results, the log-likelihood had a correlation of 0.285 at sentence level and 0.481 at speaker level (30 sentences). Franco et al. reported 0.33 and 0.5 for sentence and speaker level, respectively, but, when compared with other methods evaluated in that paper, the log-likelihood method had a low level of correlation.

In the work of Ronen et al. ([45]), both native and non-native pronunciations were rated by humans. The correlation for the log-likelihood method on the native corpus was 0.29 and 0.43 for sentence and speaker level, respectively, while the correlation on the non-native corpus was 0.06 and 0.08 for sentence and speaker level, respectively. A linearly weighted combination of the log-likelihoods for native and non-native speakers obtained a correlation of 0.44 and 0.72, respectively. All that using the same clean corpus of [36] and [13].

Neumeyer et al. ([36]) report a correlation of 0.45 at sentence level and 0.85 at speaker level (30 sentences) with methods based on duration. Their corpus consisted mostly in read speech recorded in quiet offices with high-quality microphones.

Cucchiarinni et al. ([5]) evaluated the overall pronunciation in different conditions for the speakers. On one side, read sentences were recorded through telephone with the speakers calling from their houses. In this case, the best correlation was obtained with rate of speech as method hitting 0.75 at speaker level (10 sentences, about 1 minute of speech). The other corpus was obtained recording students separated in two groups taking part of an exam in a classroom so background noise corresponding to other speakers was part of the recordings. The best correlation was 0.5 at speaker level (58 seconds of speech for one group and 75 seconds for the other) with a method consisting on the average number of phones occurring between unfilled pauses of no less than 0.2 seconds.

Teixeira et al. ([48]) analyzed only methods based on prosody in non-natives reading sentences. It is not mentioned in what context the speakers were recorded. The best correlation values at speaker level (145 sentences) were obtained by a combination of segmental features with a value of 0.71.

Hönig et al. ([23]) evaluated only prosody on speakers in two sets. In the first one, they were asked to read aloud sentences from a screen. Where this was done is not clarified but it is possible that it was collected in a laboratory. The best correlation was 0.75 at speaker level (5 sentences) using a combination of all the explored features. The other corpus was obtained remotely asking speakers to record read sentences using a headset. In this case they report that many recordings were discarded because some speech was unusable. In this case the best correlation was also obtained by using all features and was 0.57 at speaker level (around 10 minutes per speaker).

2. METHOD

Different approaches were reviewed in the previous section. An explanation of the ASR model used to generate the pronunciation scores, the scoring methods and the performance measures used to evaluate their effectiveness are presented next. The model was developed using Kaldi ASR ([37], [38]).

2.1 Automatic Speech Recognition

As discussed previously, the most successful methods for pronunciation assessment are based on Automatic Speech Recognition systems. Different aspects have proved to be important when designing an ASR system. Hystorical developments on this area will not be discussed in detail here for not being the focus of this work. A review on the history of ASR is presented by Juang and Rabiner up to developments at year 2004 in [26].

An important breakthrough in ASR systems was the development of systems based on Hidden Markov Models (HMM) trained with Gaussian Mixture Models (GMM). This has been the state of-the-art approach for decades.

In the last decade, a new family of models has appeared related to neural networks. With the developments of new algorithms and more computing power it was possible to apply this machine learning technique on different areas with successful results and one of these areas was speech recognition. The Deep Neural Network-based models for ASR will not be further discussed here. More information on DNNs for speech recognition can be found in [22].

In ASR systems the aim is to obtain

$$W = \underset{w}{\operatorname{argmax}}\, P(w|X), \tag{2.1}$$

where $P(w|X)$ is the posterior probability for segment w given the observation X (expressed in terms of features extracted from the signal). This probability is computed through Bayes rule as follows:

$$W = \underset{w}{\operatorname{argmax}}\, \frac{P(X|w)P(w)}{P(X)} = \underset{w}{\operatorname{argmax}}\, P(X|w)P(w), \tag{2.2}$$

where $P(w)$ is the language prior given by the language model and $P(X|w)$ is given by the acoustic model.

ASR systems are mainly defined by the features they use and their acoustic and language models. These aspects about the ASR system used are described in Figure 2.1 and will be discussed in the next subsections. Afterwards, the adaptation method will be explained and finally the scheme for training and evaluating the pronunciation scoring method is presented.

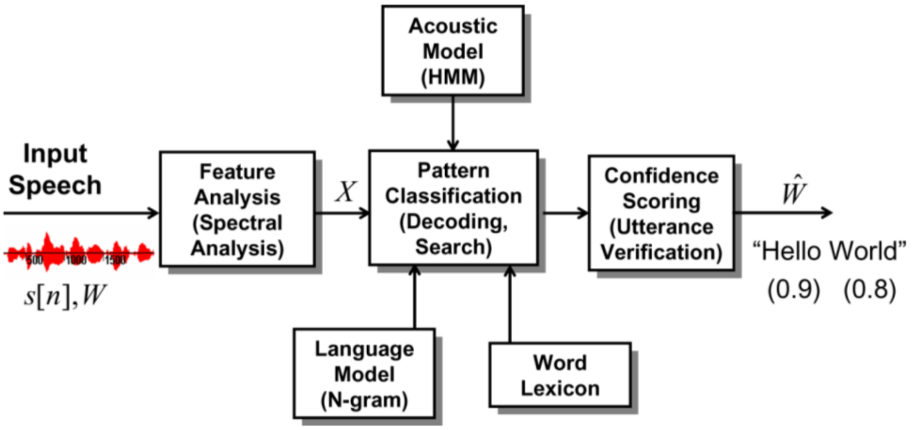


Fig. 2.1: Taken from [42]. Block diagram of an overall speech recognition system.

2.1.1 Feature Extraction

It is common practice to treat speech using a short-time analysis. This means that speech characteristics are measured in blocks of speech called frames. From each frame a set of features is computed and those serve as input for the ASR system. Two families of features are the most common nowadays: Perceptual Linear Predictive (PLP, [19]) and Mel Frequency Cepstral Coefficients (MFCC, [8]). These two have shown similar performance in ASR applications and an experimental comparison can be found in [39]. An experimental and theoretical comparison is found in [33] where the similarities between these two approaches are discussed.

Taking into account that none of the approaches was preferred over the other in the literature, Mel Frequency Cepstral Coefficients are used in this work with a standard number of features: 39 obtained from 13 cepstral coefficients plus the 13 delta (also called velocity) and 13 delta+delta (also called acceleration) features.

To obtain the named features, a series of steps are executed:

1. Sampling and Quantization: audio is a continuous signal in its nature; however, when digitalized it is comprised by a set of values determined by its resolution. A key aspect is the sample frequency, i.e. the number of samples per second. Since the main analysis of the signal is carried in the frequency domain, it is important to have a sample frequency high enough to capture the frequencies present in human voice. A sample frequency of $8000Hz$ or higher is usually adequate.

2. Preemphasis: because some high-frequency parts of speech can be atenuated by how humans produce speech or by recording elements, preemphasis compensate such atenuations and thus improves the phone recognition performance. The preemphasis is done by applying a high-pass filter on the signal, amplifying the importance of high frequencies.

3. Windowing: in order to extract information relevant for the task, speech must be analized not as one but in smaller pieces. Although audio signals are always changing, it is assumed that on short time scales they do not vary much in the information they carry. Using small portions of the signal it is possible to capture frequency components relevant for the task. The segments obtained, also called frames, can be of different sizes and can be separated by different time lags. Standard frame size and frame shift are $25ms$ and $10ms$ respectively and these are adopted for this work. It is worth noticing that a $15ms$ overlap exists between two consecutive frames and this ensures that the information between adjacent

frames is also captured in the middle of another frame. The selected size captures a small portion of the signal but big enough to contain a good number of samples. Using $8000Hz$ and frames of $25ms$ results in 200 signal samples per frame.

From the time division imposed by the frames, a windowing function is applied on the samples forming each frame. The function used is

$$(0.5 - 0.5 * \cos(2 * \pi * n/(N-1)))^{0.85}, \tag{2.3}$$

where n is the sample index within the frame and N is the number of samples in the frame. The window is multiplied by the signal for each frame to smooth the beginning and end of each frame in order to reduce discontinuities as seen in Figure 2.9.

4. Spectral Analysis: although it is possible to extract information about pronunciation directly from the waveform, the most useful analysis have been produced from a spectral analysis. The short time signal obtained from a frame is transformed into the frequency domain using the Fast Fourier Transform (FFT).

5. Mel Scale: the mel-frequency domain is more perceptually adequate to the human hearing system which discerns small changes in pitch (perceived frequency) much better at low frequencies than high frequencies. Since the purpose of an ASR system is to understand speech as humans do, this scale change from the frequency domain to the mel-scale aims to approximate the human capabilities. The formula for converting frequencies into mel scale is

$$M(f) = 1125 \ln(1 + f/700). \tag{2.4}$$

It is almost linear for frequencies up to $1000Hz$ and clearly logarithmic for frequencies higher than $1000Hz$ mimicking the human hearing system. However, the function is not applied directly on the spectral domain, a *mel filterbank* is applied instead. A set of triangular filters is applied to the computed spectral estimate. Analogously to the windowing procedure, the spectrum is multiplied by each filter and the coefficients are summed afterwards to obtain the filterbank energies. Thus, there is a resulting value for each filter. In this work 23 filters were used for being a standard amount. The positions of the filters are carefully selected in order to capture the mel scale transformation. They are non-uniformly spaced in order to have more filters in low frequencies than in high frequencies regions. An example of this layout is observed in Figure 2.2.

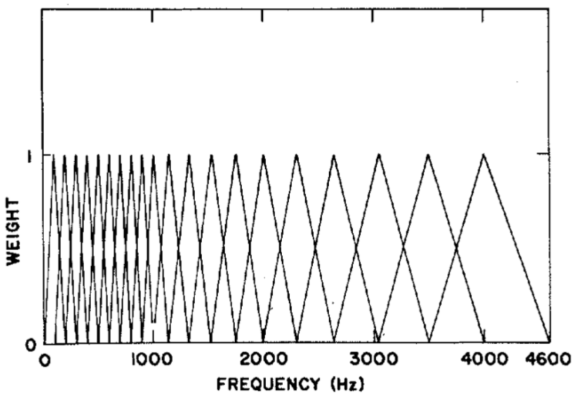


Fig. 2.2: Taken from [8]. Filter for generating mel-frequency cepstrum coefficients.

6. Filter Bank Energies: the output of each mel filter is squared and the logarithm computed over that value. This operation compresses the dynamic range of the signal mimicking human perception since the hearing system is less sensitive to differences in amplitude at high amplitudes than at low amplitudes.

7. Cepstral Coefficients: then, the Discrete Cosine Transform (DCT) is computed on the logarithms to obtain 23 cepstral coefficients. From these, only the first 12 are used in this work.

This procedure outputs 12 features. The 13th is the sum of the energy in the frame. Energy is an important feature related to phone identity and improves phone detection (for example, vowels have more energy than stop consonants). Finally, delta and delta+delta features are computed to obtain 39 MFCC features.

2.1.2 Acoustic Modelling

As previously mentioned, the acoustic model is used to compute $P(X|w)$ with X the sequence of feature vectors extracted as described above and w the sequence of words. The issues related to obtaining this probability are discussed next.

2.1.2.1 Hidden Markov Models

The acoustic model has the important duty of relating the language with the frames' features. Acoustic models in modern ASR systems are given by Hidden Markov Models (HMM).

HMMs are statistical models that consist in a set of states and transitions that connect them. None of the states are visible and each one has a probability distribution over the possible output values. Thus, the output observations depend on the states.

HMMs are particularly useful for time-dependent signals, as speech, because they assume that the system is in one state at each time instant. The transitions between the different states are expressed by a transition probability distribution and an initial state distribution expresses the probabilities of starting in each state. Then, for a speech signal, one state is the initial state and a transition from the current state to one of the connected states (eventually the same one) is produced every time step. When entering on a state, an observation is generated using the probability distribution associated to that state. Thus, states are only conditioned on the previous state and observations are only conditioned on the state that generated it. For convenience, HMMs are usually noted as $\lambda = (A, B, \pi)$, where A is the state transition probability distribution, B are the probability distributions on each state and π is the initial state distribution. More details can be found in [41] and [15].

There are many possible configurations for the HMMs. In ASR systems, the usual configuration is the *left-right* in which the only previously visited state allowed is the same state. Each state represents a particle of speech such as a word or a phone and the generated values are the MFCC features. During the recognition phase, the aim is to obtain the sequence of states that produced the observed output. However, in order to be able to produce such output, the HMM must be trained first by being presented features for which the states are known.

To train these models, it is necessary to present them with a big number of examples of each state. For ASR systems used to detect a small to medium set of words (up to around 1000 words), HMMs with states representing words are adequate but, for bigger dictionaries, it is not practical to have several utterances of each word for training. For models capable of recognizing tens of thousands of words, such as the one trained in this work, states represent phones or parts of phones. HMMs for words are formed as the concatenation of the HMMs for

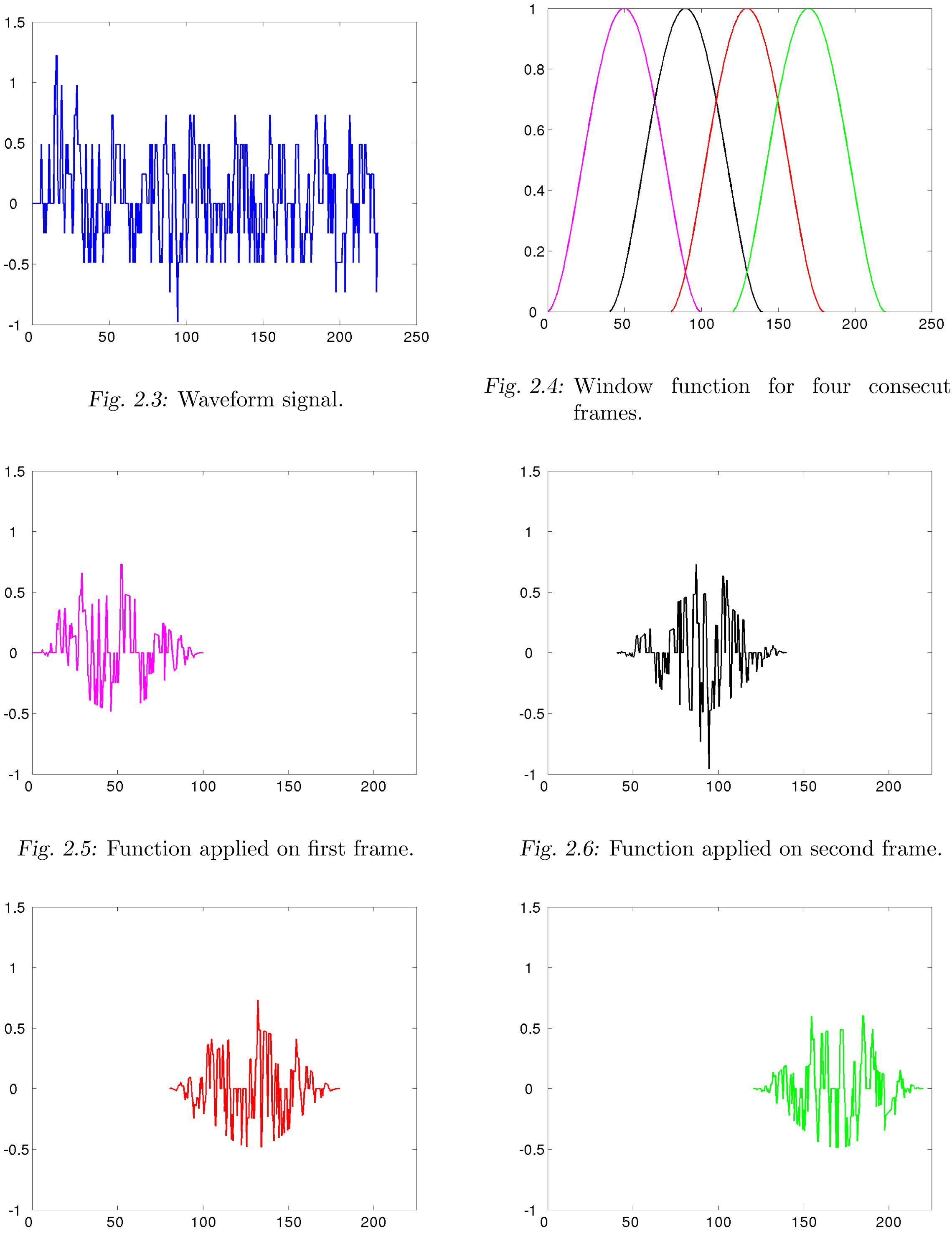


Fig. 2.3: Waveform signal.

Fig. 2.4: Window function for four consecutive frames.

Fig. 2.5: Function applied on first frame.

Fig. 2.6: Function applied on second frame.

Fig. 2.7: Function applied on third frame.

Fig. 2.8: Function applied on fourth frame.

Fig. 2.9: Windowing process on waveform signal.

the corresponding phones. HMMs for phones are usually formed by three states, *onset*, *mid* and *end* such as in Figure 2.10.

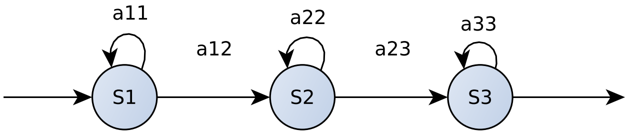


Fig. 2.10: Hidden Markov Model for a phone. State 1 is onset, State 2 is mid and State 3 is end.

Once the model is trained, given a set of observations, it is possible to know which sequence of states is most likely or even which are the k most likely sequences.

Three problems are described by Rabiner in [41], namely:

1. Problem 1: given the observation sequence X and the model λ, how can the probability of the observation given the model $P(X|\lambda)$ be computed?

2. Problem 2: given the observation sequence X and the model λ, how can the most likely sequence of states $Q = q_1, q_2, \ldots, q_T$ be obtained?

3. Problem 3: how can the model parameters A, B, π be adjusted to maximize $P(X|\lambda)$?

The first problem is the evaluation problem and can be thought of as how well does the model match the observation, an important step to know the most likely sequences of states. The second problem is the decoding and relates to the application of the model in ASR, to obtain the symbols corresponding to the features. Finally, the third problem is related to the model training.

The evaluation problem can be addressed by the forward algorithm and the decoding problem can be addressed by the Viterbi decoding algorithm as explained in [41]. For the training problem, the Baum-Welch algorithm (the expectation-maximization algorithm applied to HMMs) also called forward-backward algorithm can be used as explained in [41]. However, the Viterbi training [32] (not to be confused with the Viterbi decoding algorithm) obtains similar results faster. The distadvantage is that this algorithm does not ensure the same accuracy as the Baum-Welch algorithm but in practice the results are comparable and thus, this algorithm is used in Kaldi. All these algorithms can run efficiently by means of dynamic programming.

2.1.2.2 Gaussian Mixture Models

Each state in an HMM has an associated pdf which describes the probability distribution of the features given the state. These pdfs are usually expressed as finite sums of weighted Gaussian functions, and, therefore, are named *Gaussian Mixture Models.*

To sum up, HMMs that represent phonemes are used for the acoustic model and they capture the features information through GMMs.

2.1.2.3 Triphone models

The HMMs in state-of-the-art ASR systems generally model context-dependent phones, called triphones. Thus, instead of having a single HMM for the phoneme /p/, several HMMs are used depending on the context such as /ɪpu/, /ɪpɛ/ and so on. Therefore, an alphabet of 42 phonemes corresponds to 42 monophone HMMs or $42^3 = 74088$ triphone HMMs. Yet, monophone GMMs need to be more complex in order to capture the information about different contexts. Triphone models, on the other hand, can be less complex and use GMM with less components. This compensates, in part, the increase in the number of HMMs.

A problem with this approach is that not every triphone will have enough samples to train a separate model. In order to cope with this issue, different approaches have been proposed. One of them used pronunciation rules proposed by linguists regarding similar pronunciations between contexts. Those states with similar contexts are tied in order to reduce the number of models and to have more training data for each one. However, better results were obtained by tying states using data driven rules. The tying-states process is shown in Figure 2.11.

In step 1, a 3-state left-right HMM monophone model with a single-Gaussian pdf is created and trained for the phoneme /iː/. In step 2, for each context (t+iː+n, t+iː+ŋ, f+iː+l, s+iː+l, and so on), the triphone models are obtained by cloning the monophone ones and training the untied context dependent models using Baum-Welch re-estimation. In step 3, for each set of triphones derived from the same monophone, the corresponding states are clustered into N groups of states called *senones.* Finally, in step 4, the number of mixture components in each senone is incremented and the models reestimated until a performance measure peaks or the maximum number of mixture components is reached.

Different data driven approaches to cluster states have been proposed but one of the most used is based on a decision tree approach first introduced by Young et al. [56]. A phonetic tree is a binary tree with a question related to the phonetic context to the immediate left or right in each node. One tree is built for each state of each phone in order to cluster that state for all triphones related to that phone. Questions are selected in order to maximize the likelihood in comparison to the monophone model but ensuring that enough data is available in each senone in order to estimate the parameters of a GMM. The method works in a general way by consulting membership to phonemes sets without regard to specific classes but analyzing all subsets. Likelihoods are easily reestimated by using the means and variances of the states so the method can run quickly. This tree-based method for tying states is used for this work. See [56] for details on tree pruning and clustering.

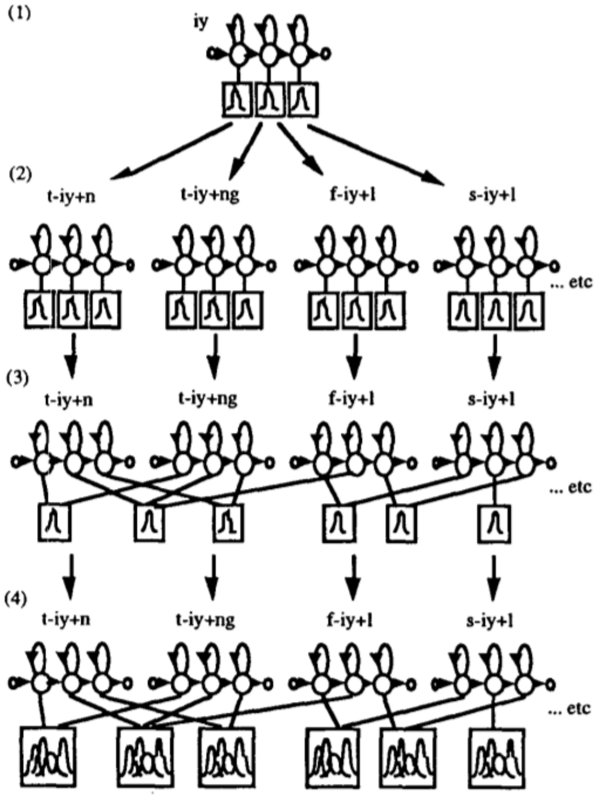


Fig. 2.11: Taken from [56]. The tied-state HMM build procedure.

2.1.2.4 Forced alignment

In some training corpora, labels are expressed at phone level. This means that the start and end time of each phone is given as input to the ASR system for training along with the features. However, the process of manually labelling phones is a very time-consuming task so, usually, labels are given at word or sentence level. In this cases, a procedure to know which phone is pronounced in each frame is needed as part of the training process.

The forced alignment problem refers to obtaining the start and end time of each phone given a string of phones such as a word or a sentence. To do so, the transcript and the features are used with the Viterbi algorithm to obtain the single best sequence of states, also called path. This path is the sequence of phones corresponding to the sequence of features.

Since the corpora used for this work have transcripts at sentence level, the forced alignment is necessary to train the senones in the context of ASR. However, for this work, it is also necessary to obtain the log-likelihoods for the methods as is discussed in 2.2. To do so, the transcripts of the speakers to be evaluated are forced aligned together with the features in their speech. Thus, a transcription at phone level is obtained together with the corresponding log-likelihoods.

2.1.3 Language Modelling

The language model in a ASR system has the purpose of assigning probabilities to sequences of words according to the likelihood of that sequence ocurring in the recognition domain. Some of the techniques to build language models consist in statistical learning from text databases, learning from grammar rules or enumerating by hand all valid strings and assigning them appropriate probability scores. One of the most common approaches is to use a statistical N-gram word model estimated from a large set of example sentences. The assumption of N-gram language models is that the probability of a word in a sentence is dependent on only the previous $N-1$ words. This probability, $P(w_n|w_{n-1}, w_{n-2,...,w_{n-N+1}})$, is estimated by counting the number of times the sequence $w_n, w_{n-1}, w_{n-2,...,w_{n-N+1}}$ appears in the language training set divided by the number of times the sequence $w_{n-1}, w_{n-2,...,w_{n-N+1}}$ appears.

In this work, trigrams were used (a 3-gram model) since it is a standard strategy for language modelling and captures a good information about the language with a relatively simple model.

Related to the language model is the lexicon, the mapping between words and phones. Since some words do not have only one canonical pronunciation, the language model should allow different phone strings for the words. Thus, in this work, the lexicon had information about the phone transcriptions of words.

2.1.4 Model Adaptation

Taking into account that a small amount of data is available to generate the non-native model, the approach to obtain an English model of non-native speakers was to adapt an English model trained with native speakers to the set of non-native speakers.

Adaptation is used as a way to reduce the mismatch between different speech conditions (e.g. acoustic conditions of the corpus), adapt to individual speakers, or specialize models (to certain population of speakers, according to sex, age, accent or any demographic population). Different adaptation methods exist and they are used for different applications. Among those in which the objective is to adapt the acoustic parameters of the model to attain a better match with the observed data, the two most used are maximum a posteriori (MAP, [31]) and maximum likelihood linear regression (MLLR, [16]).

The main advantage of the MLLR adaptation is that it does not need much data (a few seconds per speaker) to adapt the models. In order to do that, the method ties gaussians to adapt them simultaneously with the same transformation. The level of tying is, however, defined by the available data.

The MAP method uses a bayesian approach. If λ are the model parameters and x is the observation. The MAP estimate is given by

$$\arg\max_{\lambda} P(\lambda|x) = \arg\max_{\lambda} P(x|\lambda)P_0(\lambda), \tag{2.5}$$

where $P_0(\lambda)$ is the prior for the model and $P(x|\lambda)$ is the likelihood of the features given the model. In contrast, MLLR maximizes this likelihood term.

The MAP adaptation parameters are updated for those Gaussians for which samples are found in the adaption data set. For a small amount of data the disadvantage is that a small number of gaussians can be adapted and thus the model may not change much. On the other hand, with more data available it is possible for the method to update a larger amount of parameters.

In the case of speaker adaptation, it is common to have only a few seconds for adaptation. However, as will be discussed next, the amount of data available for the native and non-native classes is in the order of hours, so MAP adaptation is used in this work.

The MAP adaptation used in this work changes the gaussians' means over all HMM states appearing in the adaptation data set. The adapted mean of the GMM component k of state i, is given by

$$\widetilde{\mu}_{ik} = \frac{\tau.\mu_{ik} + \sum\limits_{t=1}^{T} c_{ikt} x_t}{\tau + \sum\limits_{t=1}^{T} c_{ikt}}, \tag{2.6}$$

where τ is a hyperparameter that measures the "strength" of belief in the prior, c_{ikt} is the probability of the mixture component k in state i given observation x_t and μ_{ik} is the mean of the original model.

If $\gamma = \sum\limits_{t=1}^{T} c_{ikt}$, then

$$\widetilde{\mu}_{ik} = \frac{\tau.\mu_{ik} + \sum\limits_{t=1}^{T} c_{ikt} x_t}{\tau + \gamma} = \frac{\tau.\mu_{ik} + \gamma \sum\limits_{t=1}^{T} \frac{c_{ikt} x_t}{\gamma}}{\tau + \gamma} = \frac{\tau}{\tau + \gamma}\mu_{ik} + \frac{\gamma}{\tau + \gamma} \sum_{t=1}^{T} \frac{c_{ikt} x_t}{\gamma} \tag{2.7}$$

and if $\alpha = \frac{\tau}{\tau+\gamma}$, then

$$\widetilde{\mu}_{ik} = \alpha\mu_{ik} + (1 - \alpha) \sum_{t=1}^{T} \frac{c_{ikt} x_t}{\gamma}. \tag{2.8}$$

One important aspect is how τ influences the adaptation. The "old" model is represented by μ_{ik}, while the second term corresponds to the "new" mean, obtained as the weighted mean of the samples found to belong to the k^{th} Gaussian with a non-zero probability. The adapted mean is then the convex combination of the old and new means. The weight is given by τ and the amount of samples that correspond to the updated Gaussian.

2.2 Pronunciation Scoring Methods

Two different scoring methods were analyzed in this work. Explanation of both of them are presented in this subsection. Besides, different approaches to obtain the log-likelihoods were analyzed along with a heuristic proposed to counteract forced alignment errors.

2.2.1 Log-Likelihood Method

This method was used in some of the first works in pronunciation assessment ([36], [13] and [29]) and is given by the likelihood obtained after doing forced alignment using an HMM model trained with native speech. This likelihood is a measure of how similar is the pronunciation in that utterance to native pronunciation.

The HMM model allows us to compute frame-level log-likelihoods: one value for each observation vector. For pronunciation scoring, though, we need scores at higher levels: phones, segments and audio. To get these scores, first, phone-level log-likelihoods are obtained from frame-level log-likelihoods. Then, segment-level log-likelihoods are obtained from the phone-level ones. Finally, audio-level log-likelihoods are obtained using the segment-level log-likelihoods.

To obtain phone-level log-likelihoods, a straightforward approach is to average frame-level log-likelihoods among each phone's frames as done in [36], [13] and [29]. This approach has the advantage that equal importance is given to every phone regardless of its length. For example, vowels are usually longer than consonants so normalizing the phone likelihoods per duration as done with the average prevents longer phones from dominating the log-likelihood at segment and audio levels. This approach is the "default" approach to obtain phone-level log-likelihoods in this work and is used in all cases except when compared with the other approaches in 4.1.3.

In a recent work by van Leeuwen and van Doremalen, [30], they studied the probabilistic properties of phone likelihoods. They studied three approaches to go from frame-level log-likelihoods to phone-level log-likelihoods: the sum, the mean and the mean multiplied by the logarithm of the duration of the phone as in Eq. 2.9, 2.10 and 2.11 respectively, where l_p is the log-likelihood for the phone p, f_0 is the frame where the phone starts, d is the duration in frames of the phone and l_f is the log-likelihood for the frame f.

$$l_p = \sum_{f=f_0}^{f_0+d-1} l_f \tag{2.9}$$

$$l_p = \frac{1}{d}\sum_{f=f_0}^{f_0+d-1} l_f \tag{2.10}$$

$$l_p = \frac{\log(d)}{d}\sum_{f=f_0}^{f_0+d-1} l_f \tag{2.11}$$

The third approach was claimed to appreciate that phones of longer duration have more acoustic evidence and should therefore induce a larger variation of log-likelihoods.

Following this idea, the sum and the log-duration scaled average were also evaluated in this work. In order to calculate the log-duration, the natural logarithm of the number of frames in the phone was obtained and multiplied by the average of frames' log-likelihoods.

Phone-level log-likelihoods are averaged over the segment to obtain segment-level log-likelihoods. This is done to give the same importance to every phone when computing the log-likelihood at segment-level. The reason for this is that, a priori, there should not be any preference on specific phones to assess pronunciation. Audio-level log-likelihoods were obtained by averaging the segment-level log-likelihoods.

2.2.2 Log-Likelihood Ratio Method

Given the native and non-native models, the Log-Likelihood Ratio (LLR) was computed using the log-likelihood of both models as depicted in Figure 2.12. The likelihood of an utterance given a model represents how likely is the utterance for that model. Hence, the difference between the log-likelihood for the native and the non-native models gives information about what model is "closer" to the utterance. Small LLR values correspond to "more native" pronunciations, while larger LLR values correspond to "more non-native" pronunciations.

2.2.3 Mitigating ASR errors

While doing forced alignments is easier than doing full ASR (since transcriptions are given, the task is greatly simplified to just getting start and end times of the phones), it is still possible for errors to appear in the alignment process. A heuristic is devised to mitigate some of the most obvious alignment errors. In our forced alignment output, some phones with 200 frames (2 seconds) were observed which were clearly mistakes. In order to reduce the impact of these errors, those phones with durations longer than expected were discarded. It is worth mentioning

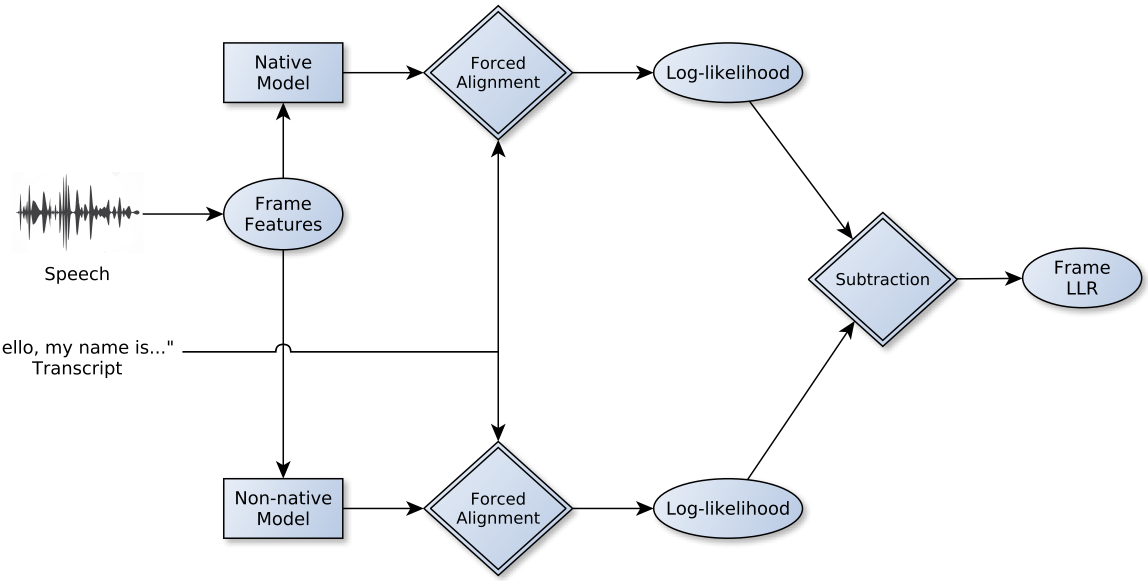


Fig. 2.12: Block diagram for LLR computation. The features and the transcript are used to do the forced alignment with each model to obtain a log-likelihood for each model. Then, the log-likelihoods are subtracted to obtain the LLR score.

that when a phone is extremely long this means that another phone duration is shorter than it should be. Only the long phones were discarded since those are clear mistakes, while it is not possible to know how many or which of the surrounding phones are shortened as a consequence of the forced alignment errors.

Different thresholds for discarding a phone are used because average durations are phone-dependent. To this end, the native speech was forced aligned using the native model. Then, the length of each phone in number of frames was calculated and the 95-percentile for each phoneme computed. Phones with durations longer than this percentile are considered likely errors and discarded when computing the LLR. Given the assumption that non-native speakers may speak slower than native speakers, the upper bounds for the length of phones were multiplied by different numbers. Four possibilities were evaluated: 1 to use the native speakers' limits, 1.5 to introduce some margin assuming non-native speakers speak slower and 2 and 2.5 to use a bigger margin.

The length of this percentile per phoneme can be seen in Table 2.1 all in number of frames. Frames are taken every 10ms so the time in miliseconds can easily be calculated multiplying the number of frames by 10ms. Significant differences in length exist among the different phones. Vocalic sounds are among the longest phones except for short vowels.

Phoneme	Duration	Phoneme	Duration	Phoneme	Duration
ð (this)	10	p (pet)	15	tʃ (check)	20
d (day)	11	n̩ (button)	16	ɝ (turn)	21
b (bad)	12	k (cat)	16	uː (food)	21
v (voice)	12	w (wet)	16	iː (see)	22
θ (think)	13	dʒ (just)	17	eɪ (eight)	23
ə (away)	13	f (find)	17	ɔ (call)	24
ɪ (hit)	13	l (leg)	17	ɫ̩ (bottle)	24
ɡ (give)	13	r (red)	17	aɪ (five)	25
t (tea)	13	z (zoo)	17	ɑr (large)	25
ʒ (measure)	14	ŋ (singer)	18	æ (cat)	27
ʊ (put)	14	m (man)	18	oʊ (go)	28
h (how)	14	s (sun)	18	ʌ (cup)	28
n (no)	14	j (yes)	18	aʊ (out)	31
ɛ (dress)	15	ʃ (she)	19	ɔɪ (boy)	31

Tab. 2.1: Duration limits per phoneme in frames. 95 percentile for all phones: 19 frames.

Related to the idea that non-natives speak at a slower pace than natives, rate of speech was computed as another scoring method based on prosodic aspects. The number of phones produced per unit of time was calculated for each audio. However, there was not enough difference between natives and non-natives in order to separate them in two clear classes. Therefore, these results will not be discussed. A hypothesis for the rate of speech not to perform well is that all non-native speakers were living in an English speaking country. Therefore, they were used to speak English quite fluently.

2.3 Performance Measures

Different measures were used to evaluate the results of the scoring methods, with different objectives. Some of them (receiver operating characteristic curves and area under the curve) were used to evaluate the binary classification capabilities of the methods while others (normalized histograms, scatterplots and Pearson correlation) were used to evaluate the classification with the multiple classes described in 3.1.3.

2.3.1 Normalized Histograms

In order to compare the distributions of each class of speakers in the one-dimensional space where the LLR lies, histograms of the distribution of the classes were generated. In this, the aim is to separate data in bins, count the instances within each bin, and then draw the function that goes through these values. The bin counts are normalized to sum to 1 and allow comparison between curves formed by different amounts of instances. Then, different curves are generated for each class to compare their distributions. An example with two classes is observed in Figure 2.13.

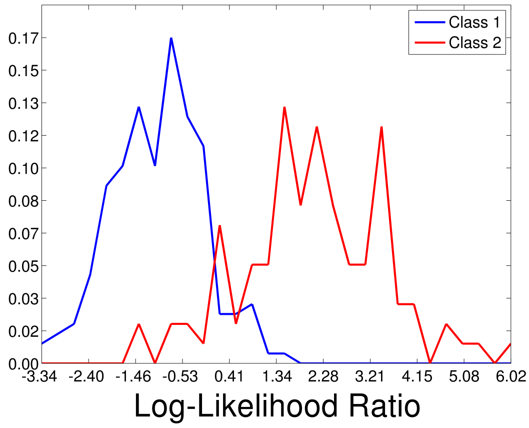


Fig. 2.13: Example of a normalized histogram.

2.3.2 Receiver Operating Characteristic curves and Area Under Curve

The Receiver Operating Characteristic curve (ROC curve) plots the performance of a binary classifier varying a discrimination threshold. The curve shows the false positive rate:

$$FPR = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \tag{2.12}$$

against true positive rate:

$$TPR = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.13}$$

for different threshold values.

The ROC curve is computed by varying the threshold in the set of LLRs obtained for natives and non-natives. As will be discussed in 3.1.3, different labels were assigned among the non-native audios. The lowest quality of pronunciation was used to train the non-native model and this class of speakers was also used to generate the ROC curves together with the natives. That is, the binary task considered when plotting the ROCs is to classify native versus non-native pronunciation of the lowest rating.

In Figure 2.14, a histogram and a threshold (marked with a green line) can be seen. With the instances of each class before and after the threshold are computed the TPR and FPR. When the threshold is swept from $-\infty$ to ∞, all the possible pairs of TPR and FPR are obtained. The ROC curve is made of all those pairs, as seen in Figure 2.15. In that curve, a green diamond shows the point corresponding to the threshold from Figure 2.14.

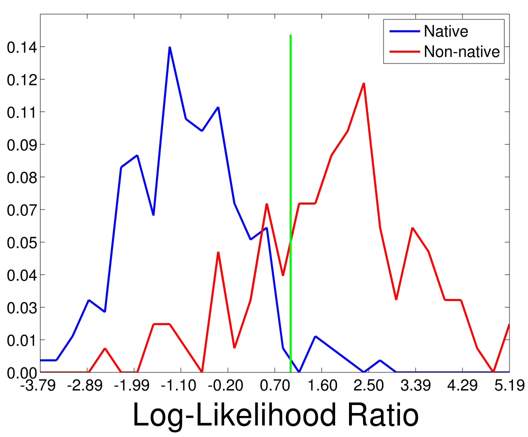


Fig. 2.14: Example of a normalized histogram with a threshold in 1. 193 instances of the Native class are on the left of the threshold and 7 on the right. For the other class, 26 are on the left and 74 on the right. $FPR = \frac{7}{7+193} = 0,035$ and $TPR = \frac{74}{74+26} = 0.74$.

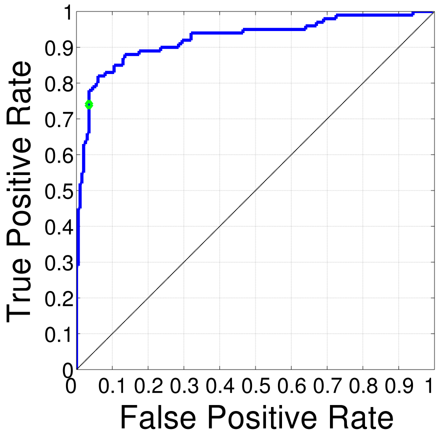


Fig. 2.15: Example of a ROC curve. Area Under the Curve is 0.923057.

Another measure related to the ROC is the Area Under the Curve (AUC). This gives an overall view of the binary classifier. It is easier to compare than ROC curves because it is only a number between 0 and 1 with values closer to 1 being better. However, in many cases it is worth analyzing the ROC curve. In this application, for example, a low FPR is preferred because that means that good pronunciation is rarely marked by the model as badly pronounced, something desirable for CALL systems.

2.3.3 Scatterplots and Pearson Correlation

Given the LLR values for the different classes it was of interest to see how good was the scoring method for predicting the pronunciation. Different ratings were manually assigned to the different classes, with 0 being native speakers, and 1 to 3 being non-natives speakers, with 3 being the worst pronunciations. Then, a scatterplot with the means of each class (using the same instances that generated the histograms) and the rating values was generated. The Pearson correlation was also computed as a measure of how well the LLR correlates with the manual ratings.

In Figure 2.16 there is an example of a histogram with all classes. The means of the instances for each class were computed in order to obtain a mean per class. Those values were plotted against the class as in Figure 2.17 to obtain the scatterplot.

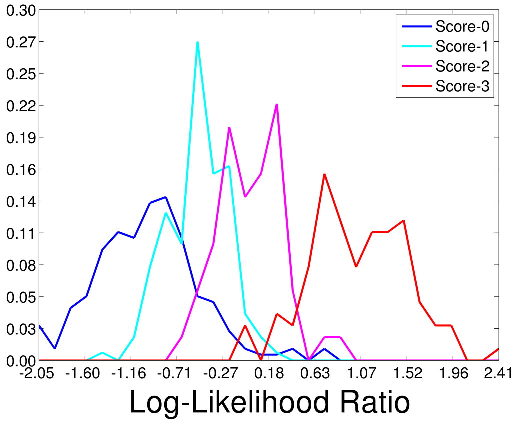


Fig. 2.16: Example of a normalized histogram with all classes.

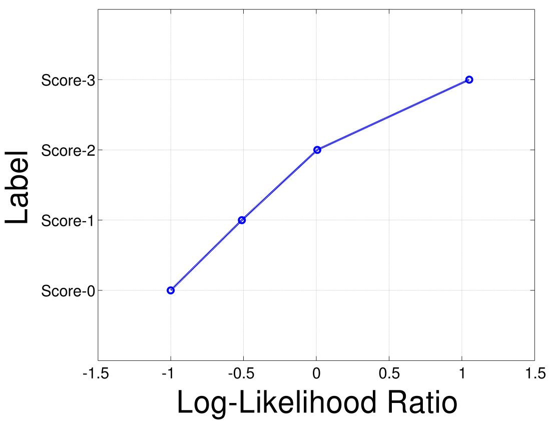


Fig. 2.17: Example of a scatterplot.

Pearson correlation was computed as follows:

$$\rho_{LLR,label} = \frac{cov(LLR, label)}{\sigma_{LLR}\sigma_{label}} \tag{2.14}$$

for the pairs instance-label in order to have a simple coefficient measuring how correlated are the LLR scores with the class labels (ratings from 0 to 3).

3. EXPERIMENTAL DESIGN

The specific characteristics of the models used are outlined followed by a description of the corpora and how they were used. Next, the assessment of non-native speakers is explained followed by an explanation of how the system was trained and the cross validation approach used to evaluate the system.

3.1 Speech Corpora

Two telephone corpora were used. The first one, *Switchboard-1 Release 2*[1] was used as training data for the ASR system. The second one, a Fisher corpus comprised two releases, *Fisher English Training Speech Part 1 Speech*[2] and *Fisher English Training Part 2, Speech*[3].

3.1.1 Switchboard Corpus

The Switchboard telephone corpus dates back to the early 1990's but since then it has suffered different reviews. 2438 conversations comprise the dataset with around 518 hours of speech. On average, each file has around 504 seconds (almost 8.5 minutes) with approximately half of that time on each side corresponding to each speaker.

The method for collecting this corpus consisted in an automatic operator system that linked a registered caller with a registered callee. Both speakers were given one of 70 discussion topics such that no two speakers conversed more than once and no one spoke more than once about the same topic. Data was collected in μlaw (encoding 14-bit signed linear PCM samples to logarithmic 8-bit samples) and with a sample rate of 8000 samples per second.

3.1.2 Fisher Corpus

The Fisher telephone corpus dates back to 2003 and was presented in two parts, the first one released in 2004 and the second one in 2005. From these, only two subsets were used in this work. One with 299 conversation sides of native English speakers speaking English and the other with 249 sides of Spanish native speakers speaking English. The first set has around 25 hours of speech while the second one almost 21 hours.

Regarding the method for collecting this corpus, although it was possible for registered speakers to call to initiate a conversation, the majority of the conversations were initiated by the automatic platform which called both speakers and linked them. Conversation topics were selected from a list and changed daily. Data was collected in μlaw (encoding 14-bit signed linear PCM samples to logarithmic 8-bit samples) and with a sample rate of 8000 samples per second.

3.1.3 Nativeness Human Ratings

In a preliminar test, the ASR system trained on the Switchboard corpus was tested on the Fisher English and Spanish natives regarding automatic recognition. The Word Error Rate (WER) was

[1] https://catalog.ldc.upenn.edu/LDC97S62
[2] https://catalog.ldc.upenn.edu/LDC2004S13
[3] https://catalog.ldc.upenn.edu/LDC2005S13

computed for three different ASR models as stated in Table 3.1[4]. Since the corpus used to train the ASR model represents an acoustic reference of nativeness, the WER for the English set was expected to be considerably lower than the Spanish one. Seeing that a considerable improvement was obtained with more complex models (triphone on a second phase) but not much difference was observed in the WER values between the two sets, a thorough analysis was carried out on the Spanish natives' set.

	English	Spanish
Monophone	73.74%	75.38%
Triphone 1	48.46%	55.00%
Triphone 2	45.47%	52.39%

Tab. 3.1: Word Error Rate for the monophone, triphone first phase and triphone second phase models on English and Spanish speakers sets.

We listened to all conversations with a side tagged as a Spanish native speaker to analyze the characteristics of the non-native set that could explain such similar WERs. Different grades of pronunciation profficiency were heard, so different labels were assigned to each speaker taking into account both phonetic and prosodic aspects. The labels were: *Score-1* for speakers with a degree of nativeness comparable to English natives, *Score-2* for speakers with excellent pronunciation but some recognizable errors, and *Score-3* for speakers with serious pronunciation mistakes but capable of having a conversation.

It should be noted that the pronunciation assessment was carried out by me, a non-native with an advanced level of English as second language. Besides, my native language is Spanish so knowledge of the expected errors for the Spanish natives was taken into account when labelling the speakers.

Of the 249 sides corresponding to Spanish natives, the proportions of each class are described in Table 3.2. Considering the number of speakers with a proficiency comparable with that of natives, the WERs obtained for the Spanish category can be explained by the fact that the majority of speakers have excellent pronunciation.

	Total	Score-1	Score-2	Score-3
Amount	249	144	46	59
Percentage	100%	57.83%	18.47%	23.69%

Tab. 3.2: Spanish nativeness categories proportions.

After separating the Spanish speakers in the three classes aforementioned, different WERs were observed as reported in Table 3.3, where clear differences appear among the classes. While those speakers labelled as *Score-1* have a WER similar to that of English natives, those speakers tagged as *Score-3* obtain much worse results than natives. *Score-2* shows WERs in between *Score-1* and *Score-3*. These results are more compatible with our expectations. The rest of the work was based on this classification. It should be noted that all English natives were labelled as *Score-0*.

[4] Kaldi authors reported %47.7 of WER on the Fisher corpus with a similar ASR model. This value is in between the results for English and Spanish speakers' results obtained here and thus agree with the expected results.

	Score-0	Score-1	Score-2	Score-3
Monophone	73.74%	73.67%	77.47%	77.57%
Triphone 1	48.46%	50.62%	57.90%	63.15%
Triphone 2	45.47%	47.72%	55.51%	60.98%

Tab. 3.3: Word Error Rate for the monophone, triphone first phase and triphone second phase models on English and Spanish categories speakers sets.

3.2 Model Training

A native model was trained using the Switchboard corpus as described in 3.1.1. The language model was based on an trigram trained over the sentences in this corpus. The lexicon contained partial words, words containing laughter, common alternate pronunciations of words, hesitation sounds, proper nouns, anomalous words, coinages, and normal lexical items[5].

For this corpus the phoneset was comprised by not only phonemes but also noise and laughter among other non-phoneme symbols. Moreover, for each phoneme four categories are distinguished in the lexicon. Thus, for /b/ there are four phones `b_B b_E b_I b_S`. The “`_B`” corresponded to the phone used as *beginning* of word, “`_E`” when in the *end*, “`_I`” for appearances in the middle (*in*) of a word and “`_S`” for interrupted words where only the first part of the word was pronounced. This totaled 199 phone symbols from which 168 were from phones comprising 42 English phonemes. For the phone analysis carried in the experiments, the four categories for each phone were merged into one.

Initially, a monophone model was trained with a total of 146 HMMs comprising 989 gaussians. The difference between the number of HMMs and the number of phones in the lexicon (199, as explained above) is due to some of the phones not being present in the corpus and some of them being tied together by the tree-based tying method (which is done as for the senone-based models, even for the monophone case).

The monophone model was used as input for the triphone model in order to align the triphones. The resulting model had 2344 senones and 30087 gaussians. Finally, in order to obtain a more complex model, a second triphone model was trained using the alignments from the first one. The final model had 2972 senones and 70125 gaussians.

Once the ASR model was trained on the Switchboard corpus, the native and non-native models had to be defined. One option for the native model was to use the ASR model trained on the Switchboard corpus. However, since all development data is part of the Fisher corpus, some acoustic mismatch may exist between the two corpora due to different acoustic conditions. Although both of them were collected in telephone conversations, different technologies in the telephones may exist since the two corpora were collected separated by almost a decade. The acoustic mismatch may influence the recognition process. Hence, ideally the model used to generate the scores should be trained with data in the same acoustic conditions and ideally from the same collection. This was not possible since the Fisher corpus is not big enough to train the ASR model from scratch.

Hence, in order to cope with the mismatch, three approaches were explored. The default scheme used as native model the one adapted to Fisher native speakers' data and the non-native model was obtained by adapting the native model adapted to Fisher with non-native data. This scheme, referred as *Scheme 1*, is used in all cases except when compared with the other two in 4.1.4. The diagram of this model can be seen in Figure 3.1.

[5] Obtained from http://www.openslr.org/resources/5/switchboard_word_alignments.tar.gz or http://www.isip.piconepress.com/projects/switchboard/releases/switchboard_word_alignments.tar.gz

The *Scheme 2* obtains the native model by adapting the ASR model with Fisher natives as in Scheme 1 but the non-native model is obtained by adapting the ASR model directly with the non-natives. This was evaluated to see the effects of adapting both acoustic mismatch and speakers' characteristics. The diagram of this model can be seen in Figure 3.2.

Finally, the *Scheme 3* obtains the non-native model by adapting the ASR model with Fisher non-natives as in Scheme 2 but the native model is the ASR system trained with the Switchboard corpus itself. This was evaluated to see the effects of the acoustic mismatch between corpora. The diagram of this model can be seen in Figure 3.3.

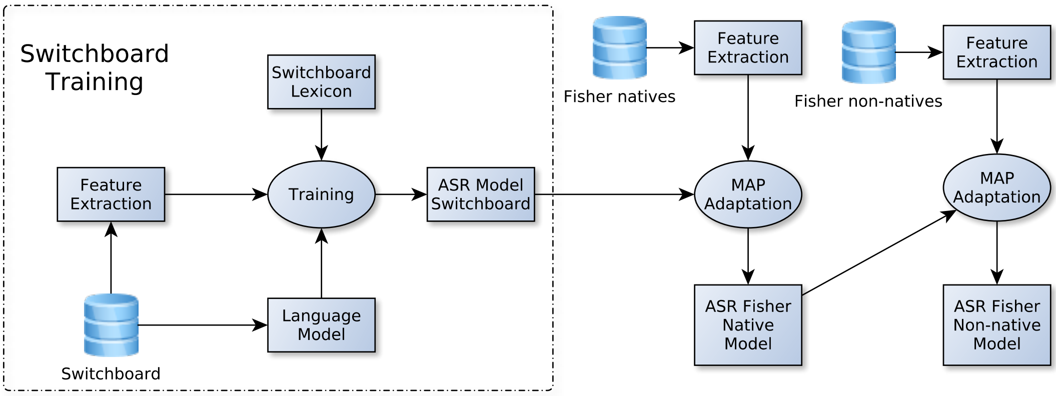


Fig. 3.1: Block diagram of adaptation Scheme 1.

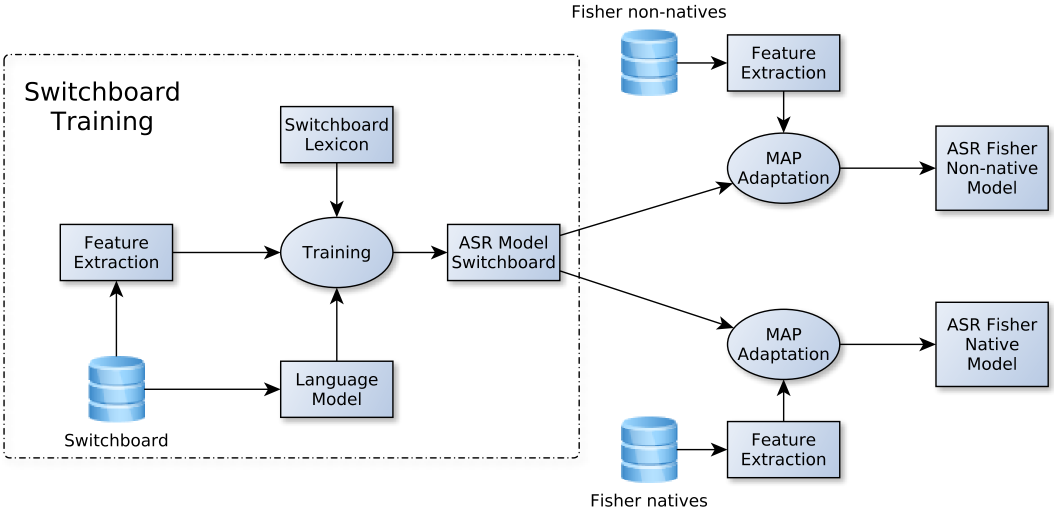


Fig. 3.2: Block diagram of adaptation Scheme 2.

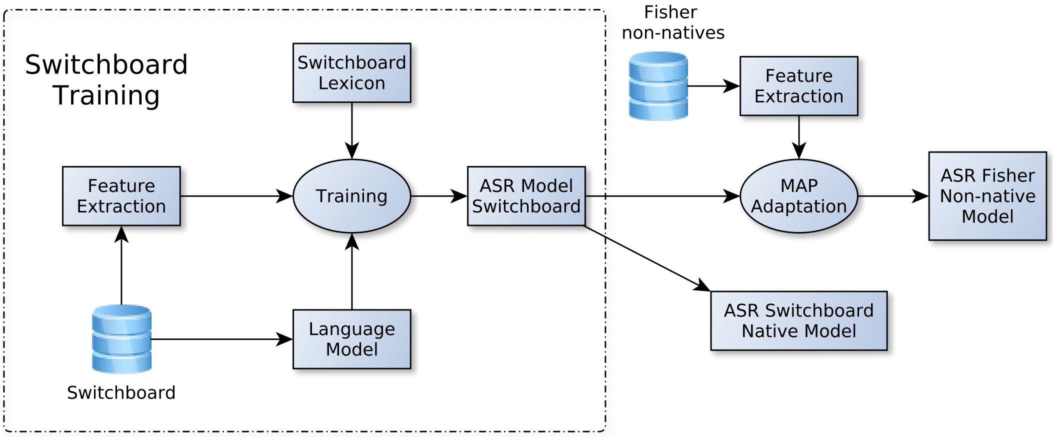


Fig. 3.3: Block diagram of adaptation Scheme 3.

3.3 Development and Test Sets Definition

For this work, the Fisher data was separated into test and development sets. This was done to keep a held-out set to see how well did the method generalize after all tuning and model selection. This tuning and model selection, as well as the training of the final system, was done on the development set. For development, k-fold cross validation was done on the development set as explained in Section 3.4.

The division into sets was carried out at speaker level. The importance of having all utterances from a speaker in the same set is that, if one speaker is present in both sets, performance on that speaker will be biased because the model is already trained using information about that specific speaker. This can lead to inaccurate performance measures on the system, similar to using the same data both for training and testing in any machine learning problem.

Before doing the division into test and development sets, a review on the number of conversations of each speaker was carried out for all speakers. The numbers of speakers in each class with different numbers of conversations are in Table 3.4.

#conversations Class	1	2	3	4	5	Total	#speakers
Score-0	231	34	0	0	0	299	263
Score-1	24	19	26	1	0	144	70
Score-2	6	8	8	0	0	46	22
Score-3	8	11	8	0	1	59	28
All	269	72	42	1	1	548	383

Tab. 3.4: Number of speakers per number of conversations and number of speakers per class.

In order to leave as test set a relevant number of conversation sides with no common speaker with the development set, all speakers with only one conversation were left as held-out set for the Spanish categories (Scores 1 to 3). With this criterion, around 30% of each class at speaker level was left as held-out, while maximizing the number of conversations left for development. It was not necessary to leave all speakers with one conversation in the English class (Score-0) for held-out since there were plenty of data for this class in comparison with the other classes and most speakers had only one conversation. The percentages of each class both for speaker and conversation level are presented in Table 3.5. It can be seen that the proportion of speakers held-out for testing is between 24% and 34% for all classes. Regarding conversations, the proportions are considerably lower given the design choices.

	Conversation		Speaker	
	Development	Held-out	Development	Held-out
Score-0	234 (78.26%)	65 (21.74%)	200 (75.47%)	65 (24.53%)
Score-1	120 (83.33%)	24 (16.67%)	46 (65.71%)	24 (34.28%)
Score-2	40 (86.95%)	6 (13.04%)	16 (72.72%)	6 (27.27%)
Score-3	51 (86.44%)	8 (13.56%)	20 (71.43%)	8 (28.57%)

Tab. 3.5: Number and percentage of development and held-out sets for speakers and conversations.

3.4 Cross Validation

K-fold cross validation was used when producing the pronunciation scores. K-fold cross validation consists in separating data in sets and then run the model training on all sets but one, testing that model in the remaining set, and repeating the process for all sets. This has the advantage of maximizing the amount of training data for each model, also avoiding biased results because of a specific train/test division. The same approach as when dividing data for test and development was used forcing all conversations of each speaker to be in the same set to avoid optimistic results.

With the intention of having all possible data for training the models, the number of folds was based on the number of speakers in the smallest set, that of *Score-3* speakers. Having 20 speakers for the development set, 20 folds were used each one with one speaker with Score-3.

Test was carried out on each fold after using the other folds to obtain adapted native and non-native models using the Score-0 and Score-3 samples. The resulting scores were then pooled to show the overall performance on the full set of development data.

4. RESULTS

The different variations for the models and methods mentioned during the method description (2.2 and 3.2) are analyzed in this section. First, different experiments were carried out on the development set to select the best choices among the explored configurations. Next, the preferred options were analyzed in the held-out set in order to obtain a final evaluation of the system on fresh data and confirm that the results were not optimistic due to overfitting during the development phase.

4.1 Results on Development Set

Different options were explored in order to obtain the best possible scoring system. First, two methods are compared for computing scores at conversation or "audio" level: log-likelihood (LLK) and log-likelihood ratio (LLR) both varying the amount of aggressiveness in the adaptation process. Then, for the best method, two smaller levels are evaluated: segment and phone in contrast with the entire conversation to obtain the best level of adaptation aggressiveness for each region level.

Afterwards, the three proposed procedures to obtain phone log-likelihoods from frame log-likelihoods are evaluated. Using the best configuration until then, the three adaptation schemes proposed are evaluated and the outcome with different amounts of data analyzed. Finally, the error mitigation heuristic is put to test.

4.1.1 Log-Likelihood versus Log-Likelihood Ratio

In order to obtain a pronunciation score, two methods were first evaluated: LLK and LLR. For both of them, different values for the adaptation hyperparameter τ explained in 2.1.4 were explored. The comparison was done by plotting ROC curves for each method and each τ value as seen in Figures 4.1 and 4.2.

A huge difference exists between the two methods, with LLK being only slightly better than a random guess. Furthermore, the level of aggressiveness in the MAP adaptation does not produce any considerable difference for this method.

On the other hand, the LLR produces good results, showing different performances for different τ values. A low value of τ means a high level of belief on the adaptation data, as seen in 2.1.4.

Considering these results, LLK was not longer explored so all results from this point on are with the LLR method.

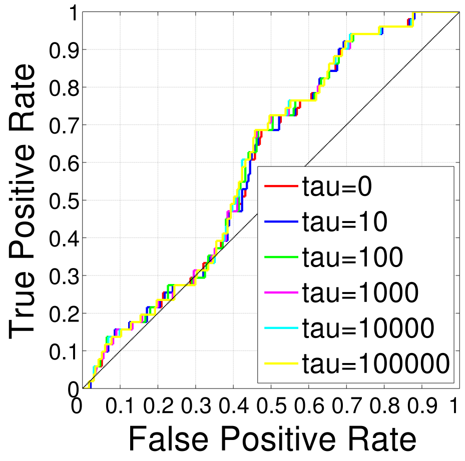


Fig. 4.1: ROC curve for LLK at audio level.

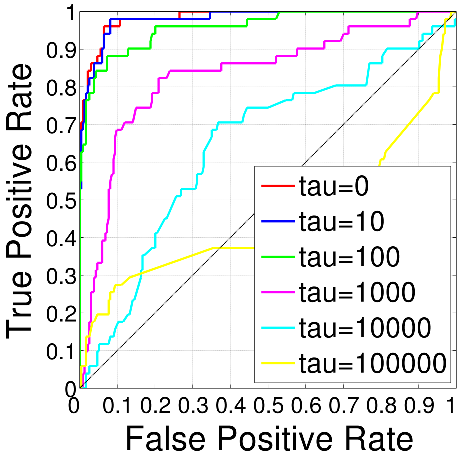


Fig. 4.2: ROC curve for LLR at audio level.

4.1.2 Different Levels of Scoring

In this subsection, the different τ values are evaluated at segment and phone level as seen in Figures 4.3 and 4.4. The area under the curve for the three levels: audio, segment and phone are reported in Table 4.1.

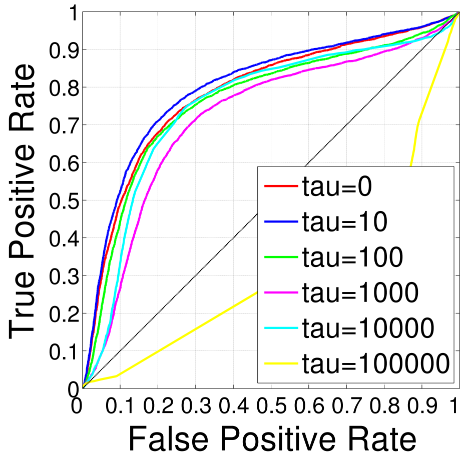


Fig. 4.3: ROC curve for LLR at segment level.

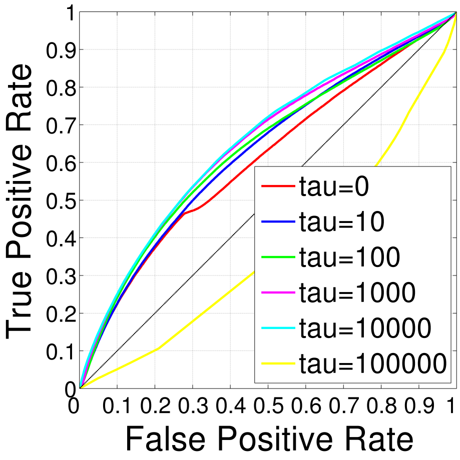


Fig. 4.4: ROC curve for LLR at phone level.

Level Tau	Audio	Segment	Phone
0	0.979	0.788	0.639
10	0.977	0.803	0.652
100	0.953	0.774	0.681
1000	0.831	0.741	0.716
10000	0.652	0.784	0.731
100000	0.575	0.165	0.263

Tab. 4.1: Area under the ROC curve per τ at Audio, Segment and Phone levels.

Significant differences are seen between the different levels with shorter levels having worse results. This is expected since the shorter the level, the less information available to judge its pronunciation. Longer regions carry more information and can reduce the influence of atypical sounds such as noise or non-canonical pronunciations of certain phones or words. Thus, phones have scores that tend to be noisier than segment-level scores which in turn are noisier than audio level scores. This is related to the fact that each phone lasts, on average, 99.6 miliseconds; segments last, on average, 3.76 seconds; while audio last, on average, 295 seconds (almost 5 minutes).

When using a low value for τ, the resulting scores have less bias and more variance as predictors of the true scores than with higher values. With higher levels, such as audio and segment, the variance of the scores is reduced since the score is computed as an average over the scores from shorter levels. Hence, it is possible to use a model with higher variance (and lower bias) which has more discriminative capabilities. On the other hand, phone level scores are averages over few frames only, which is not enough to mitigate the variance of a model adapted with a low τ value.

Consequently, depending on the level, different τ values produce the best results. For audio, both 0 and 10 are very similar with 0 showing a little better AUC, so 0 is selected. For segment, 10 has the best performance while 10000 shows the best results for phone level. For the purpose of this work, these values are selected and next experiments will use such values for each level.

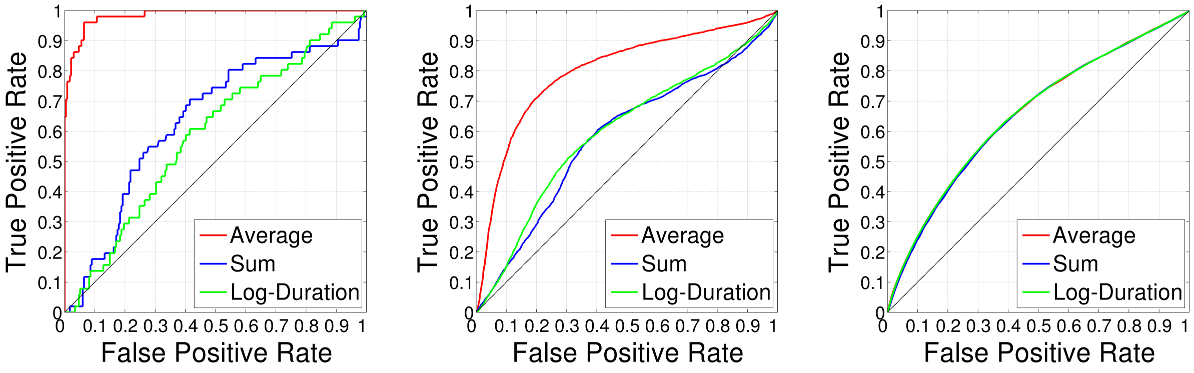
4.1.3 Transforming Frame-level to Phone-level Log-Likelihoods

As discussed in 2.2, different approaches to obtain phone-level log-likelihoods from frame-level log-likelihoods can be devised. Up to this point, only the average was used. In this subsection the sum and log-duration average are evaluated.

Figures 4.5a, 4.5b and 4.5c show the results for audio, segment and phone levels respectively. In both audio and segment the average is clearly better than sum and log-duration, while for phone-level they are all comparable.

One possible explanation for the log-duration and sum approaches performing worse than the average is that they assume that longer phones carry more acoustic information and thus must weight more in the score. However, when assessing pronunciation, each phone should contribute the same to the longer-level averages, regardless of their duration. A badly pronounced long vowel should contribute the same as a badly pronounced short consonant.

Although big differences exist in audio and segment levels, the three methods have similar results at phone level. One possible explanation for this behavior is that for phone level, the adaptation is performed with a high value of τ and thus, both native and non-native models have a strong belief in the prior-to-adaptation model. Some more experiments with other τ values would be necessary in order to have a minimal pair comparison scenario. Since the sum and log-duration approaches did not produce better results than the mean for audio and segment levels, we decided to focus in other experiments.



(a) Audio level.

(b) Segment level.

(c) Phone level.

Fig. 4.5: Comparison of frame-level log-likelihoods to phone-level log-likelihoods approaches.

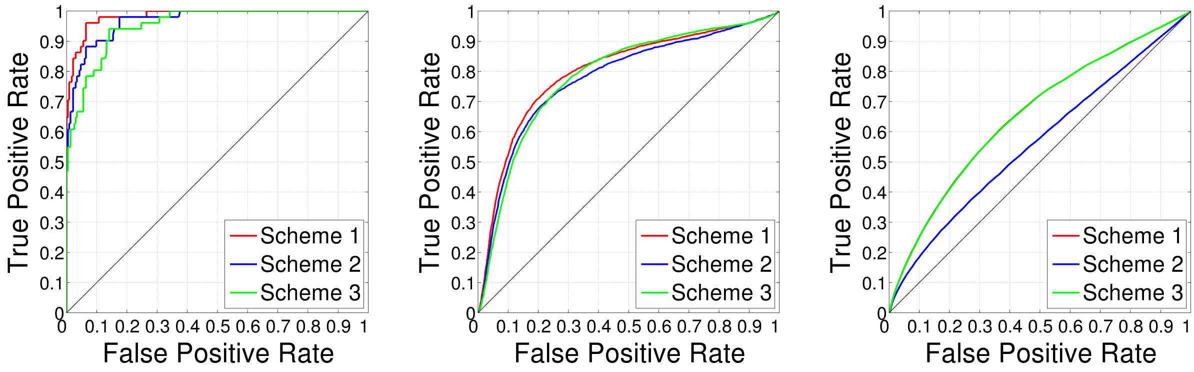
4.1.4 Adaptation Scheme Analysis

In this subsection, the three adaptation schemes are evaluated. Scheme 1 obtains the native model by adapting the Switchboard-trained model and the non-native model by adapting this native model to the non-native adaptation data. The reason for doing this is that there is a larger amount of native than non-native data. Hence, the two-step adaptation process should lead to a more robust non-native model. Scheme 2 obtains both the native and non-native models by adapting directly to the Switchboard-trained model. Scheme 3 uses as native model the Switchboard trained one and the non-native model is obtained as in Scheme 2. For more information refer to Section 3.2. Figures 4.6a, 4.6b and 4.6c show the ROC curves for audio, segment and phone level respectively. Table 4.2 shows the area under the ROC curve in order to compare similar curves.

As was expected, the Scheme 1 produced better results than the other two. However, no big differences were observed in audio nor segment levels. One possible reason for the schemes obtaining similar results is that, although different corpora were used to obtain the models, all data were telephone conversations captured with a similar setup and, thus, there was not a big acoustic mismatch between Switchboard-trained and Fisher-trained models.

However, at phone level, there is some difference between the Scheme 2 and the other two. The main difference in this case is that both Schemes 1 and 3 obtain the non-native model by adapting the native model with non-native data, while Scheme 2 obtains the non-native model by adapting a model that is not the same as the native model used to compute LLR. Hence, in Scheme 2, the native and non-native models are obtained both from a common model and it is possible that both adaptations capture the corpora mismatch more than the population mismatch and the two models end up being very similar at phone level. This happens at phone level for which the adaptation is very conservative and then, the only condition captured by the adaptation is the most evident, the acoustic mismatch. Nevertheless, for audio and segment levels, the adaptation is more aggressive and the population mismatch is not subsumed by the acoustic mismatch, then producing better results in those levels.

Being Scheme 1 the best performing one, the next experiments continued using that configuration.



(a) Audio level. *(b)* Segment level. *(c)* Phone level.

Fig. 4.6: Adaptation schemes comparison.

Level Scheme	Audio	Segment	Phone
1	0.979	0.803	0.731
2	0.961	0.782	0.602
3	0.942	0.787	0.731

Tab. 4.2: Area under the ROC curve for the three schemes at audio, segment and phone levels.

Varying the Amount of Data Available for Adaptation

Given the results in the previous experiment with different adaptation configurations, another set of experiments with different amounts of data available to adapt the models was carried out. For this experiment only Schemes 1 and 2 were considered.

Each conversation has approximately 5 minutes (300 seconds) of speech per speaker [1] in each conversation. Different lengths of speech were taken from each conversation up to 5, 10, 20, 40, 60, 80, 100, 120, 140, 160, 180, 200, 220, 240, 260 and 280 seconds.

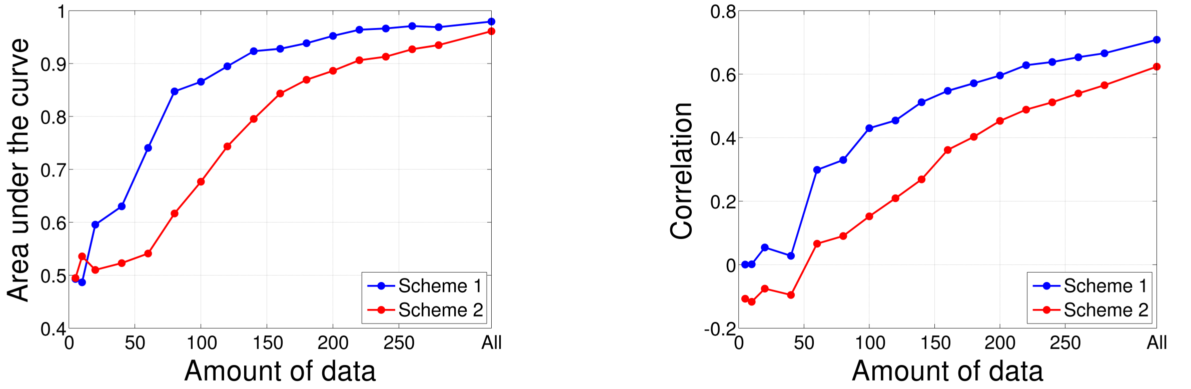
Figures 4.7, 4.8 and 4.9 show graphs for the area under the ROC curve and the correlation with human labels in terms of seconds per conversation used for adaptation for audio, segment and phone levels respectively. It is possible to see that in all three levels the Scheme 1 had better results for all amounts of data. However, the correlation at phone level was extremely low in all cases showing a low performance for that level despite the data used or the scheme.

Nevertheless, the difference between the two schemes is bigger when a low amount of data was available. This showed that the adaptation to the Fisher native speakers before the adaptation to the non-native speakers produced larger gains when data was scarce. Furthermore, in some cases, double the data was necessary for the Scheme 2 to obtain the same results than Scheme 1.

This experiment was also useful in order to estimate the amount of adaptation data necessary to obtain results similar to using all adaptation data. In comparison with using all data for adaptation (around 300 seconds per speaker per conversation), the Scheme 1 showed very similar results with around 200 seconds showing that small gains were obtained after this point. However, using all data showed better results in all cases. Thus, some improvement may be possible if more data were available showing that the peak of performance was not yet attained. This

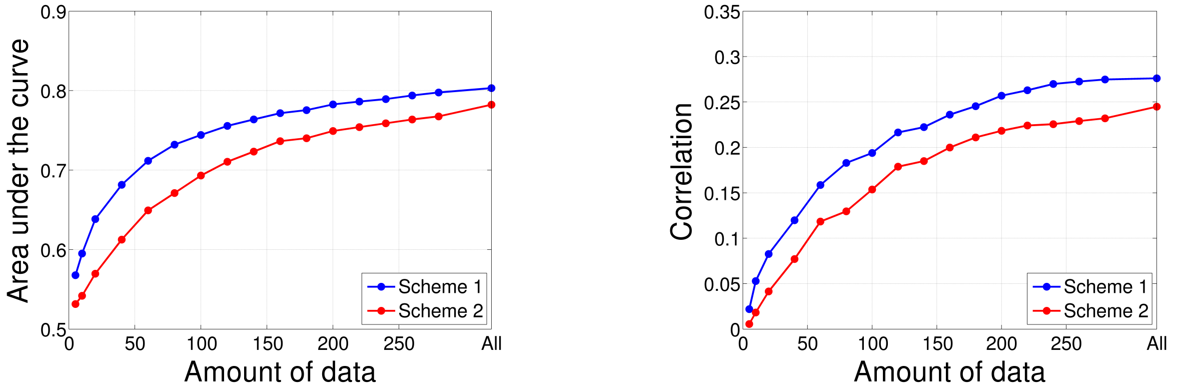
[1] 295 seconds on average per conversation per speaker.

is not the case for segment and phone level for which using 280 seconds and all data produced similar results.



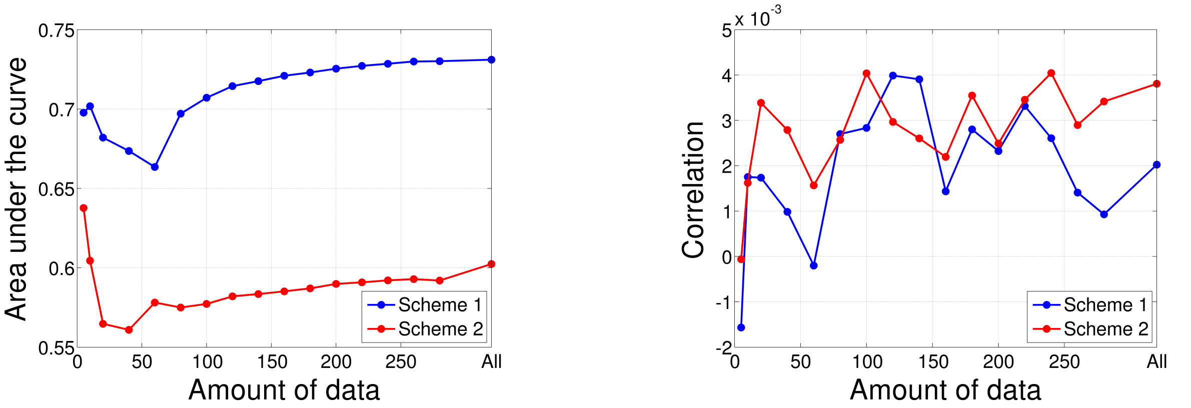
(a) Area under the ROC curves. *(b)* Pearson correlation with human labels.

Fig. 4.7: Comparison between schemes for audio level with different amounts of data.



(a) Area under the ROC curves. *(b)* Pearson correlation with human labels.

Fig. 4.8: Comparison between schemes for segment level with different amounts of data.



(a) Area under the ROC curves. *(b)* Pearson correlation with human labels.

Fig. 4.9: Comparison between schemes for phone level with different amounts of data.

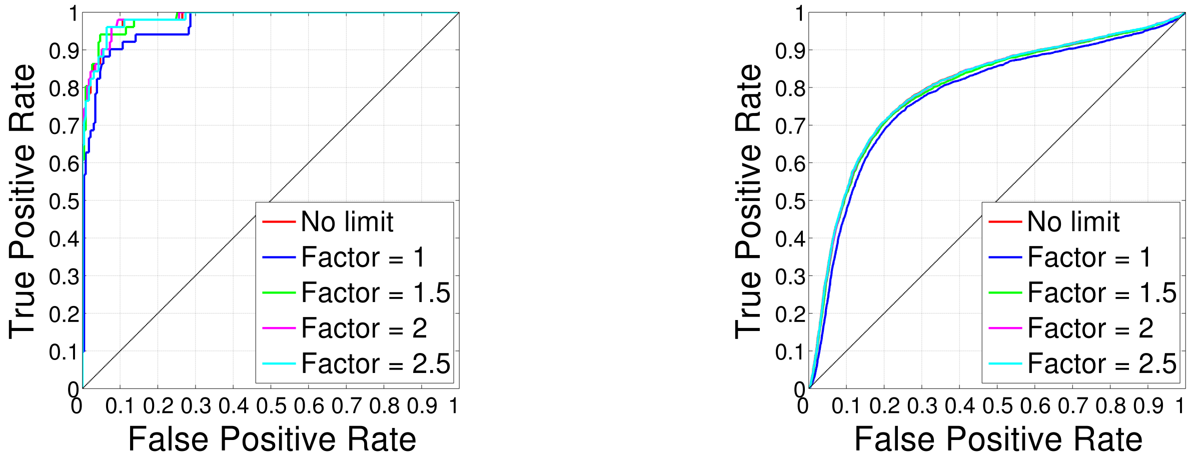
4.1.5 Heuristic for ASR Error Mitigation

As explained in 2.2.3, a limit in the duration of phones was proposed to counteract forced alignment errors. The length of the 95-percentile of each phoneme was multiplied by a factor in order to obtain a limit in the length of a phone corresponding to that phoneme. The factor 1 used the exact same duration limit computed for the native set, while 1.5, 2 and 2.5 allowed phones with more duration, assuming that non-natives may speak slower than natives. Every phone with a duration longer than the limit imposed was discarded for averaging segment and audio scores. Since phones longer than the corresponding limits were discarded, the phones used at phone level are not the same than in previous experiments. Thus, the results for phone level are not presented for being incomparable with previous results.

The "no limit" model variant, in which all phones are used, was compared with the different factor values. The "no limit" corresponds to the model used for all the previous experiments. The results can be seen in Figure 4.10.

As can be seen, the heuristic did not produce large improvements. This showed that the phones with an extremely long duration due to alignment errors did not influence the scores in a high degree.

However, it should be noted that factors greater or equal than 2 did not produce very different results from not using the heuristic. This is because the limit imposed with a factor of 2 allowed almost every phone, such as in the case when no heuristic was used.



(a) Audio level.

(b) Segment level.

Fig. 4.10: ROCs for error mitigation heuristic.

4.1.6 Multiclass Classification Results

Once the best configuration in previous experiments was selected, a more thorough analysis was carried out using all speaker classes. The best model used log-likelihood ratio and the phone log-likelihoods were obtained averaging the frames log-likelihoods. The adaptation was done with Scheme 1 and it used $\tau = 0$ at audio level, $\tau = 10$ at segment level and $\tau = 10000$ at phone level. No heuristic was used to filter phones based on their duration.

The normalized histograms with all classes are presented in Figures 4.11, 4.12 and 4.13 for audio, segment and phone levels respectively.

Figure 4.11 shows that the means of the score distributions for the different classes are ordered as expected based on their rating. When analyzing the histograms at segment (Figure 4.12) and phone (Figure 4.13) levels, the separation between the classes is less clear being the extreme case at phone level where curves overlap significantly. This goes in the line of the previously discussed results where at smaller duration levels the classification capabilities of the system diminishes. Furthermore, the dispersion in the scores grows as the level is smaller. At both segment and phone levels, the classes had longer tails on each side which were left out of the graphs to have a better view of the middle part.

In general, classes with worse pronunciation present more dispersion while better pronunciation classes such as Score-0 present a more compact shape showing that natives tend to be slightly more consistent in their scores.

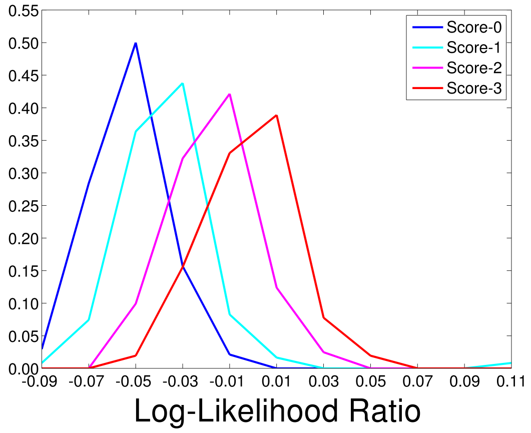


Fig. 4.11: Histogram of all classes for audio level.

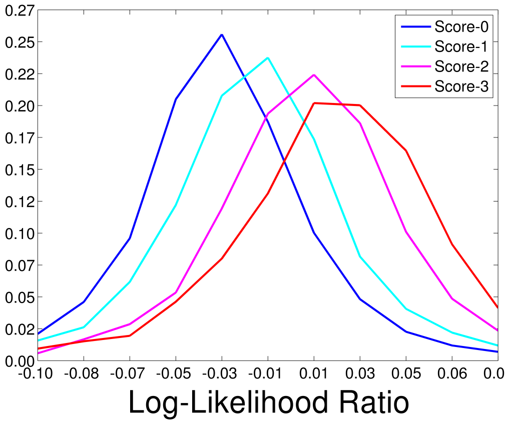


Fig. 4.12: Histogram of all classes for segment level. The 2.5% of lowest values and the 2.5% of highest values were removed for visualization.

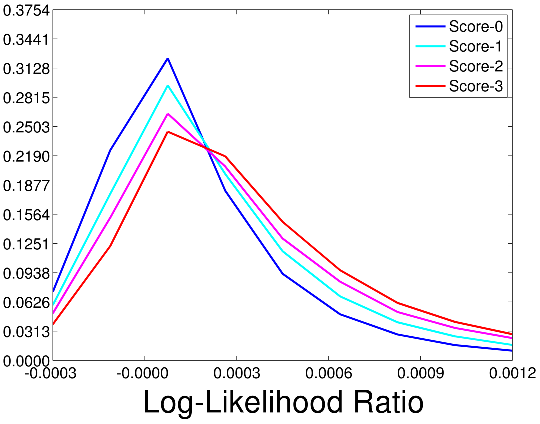


Fig. 4.13: Histogram of all classes for phone level. The 5% of lowest values and the 5% of highest values were removed for visualization.

The next step was to evaluate how well did the pronunciation score correlate with the class labels. To do that, the scores for all instances in one class were averaged and that average plotted against the label as seen in Figure 4.14. The three levels show different behaviors but all of them are close to linear. Figures 4.15, 4.16 and 4.17 show for audio, segment and phone respectively the same averages of Figure 4.14 and two triangles showing $average + standard\ deviation$ and $average - standard\ deviation$ to indicate how disperse are the scores for each class.

As with the normalized histograms it is possible to see that with phone level the score values are less separated than at segment level and in turn segment values are less separated than audio. Again, it is possible to see a slight difference in the dispersion between Score-3 and Score-0. Nevertheless, the native class (Score-0) was the only one not rated by a human since all the speakers were English natives. Part of the dispersion in the other classes could be explained by the fact that only one human rater assessed all conversations. This is one of the reasons why more than one rater is used to score pronunciation, as discussed in Section 1.2.

Pearson correlation was computed also for the three levels as shown in Table 4.3. A large level of correlation was obtained for audio level showing that the system is capable of predicting human ratings. For segment and phone level, the correlation is greatly degraded showing the importance of the amount of data for the quality of the method.

Nevertheless, another explanation for the large drop in correlation from audio to shorter levels is that the labels were assigned at audio level and those labels assumed for all segments and phones contained in them. Although it is a simple approach, it is a rough assumption because it is possible for a speaker to make mistakes with certain phones but have a correct or even almost native pronunciation in others. Thus, it is possible for part of the poor results for segment and phone levels to be explained by this assumption. However, the cost of labelling at segment or phone level is considerably more expensive than at audio level so this task remains pending.

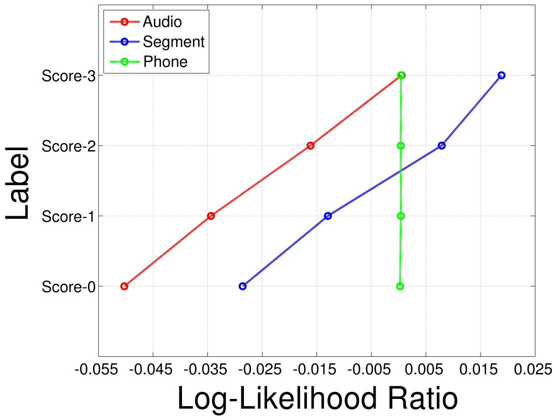


Fig. 4.14: Mean of instances for each class for each level.

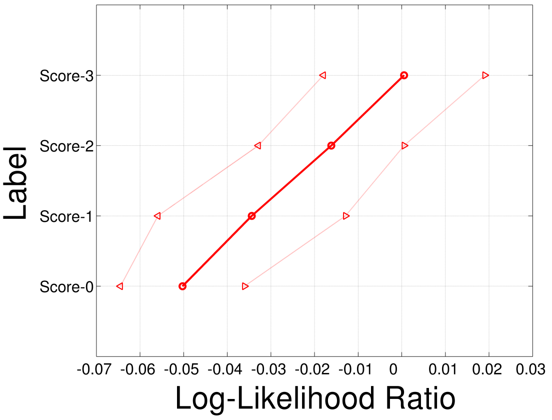


Fig. 4.15: Mean and mean ± standard deviation for audio level.

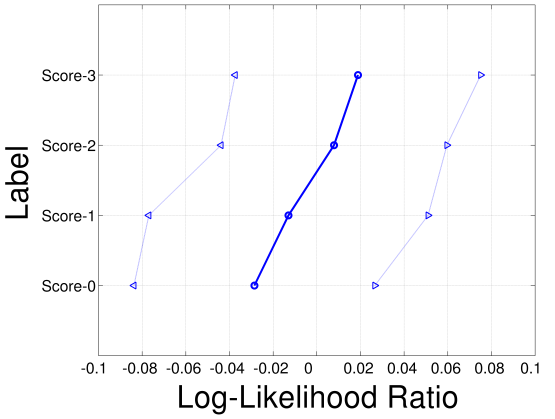


Fig. 4.16: Mean and mean ± standard deviation for segment level.

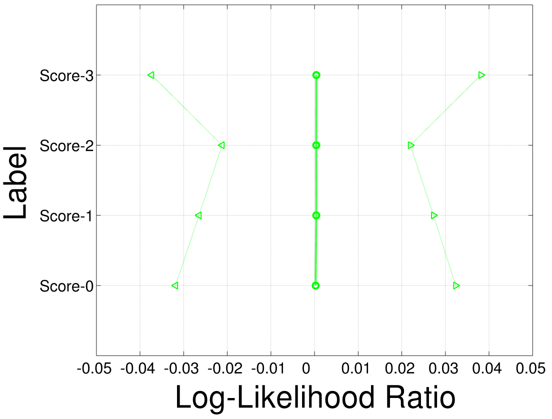


Fig. 4.17: Mean and mean ± standard deviation for phone level.

Audio	Segment	Phone
0.708	0.276	0.002

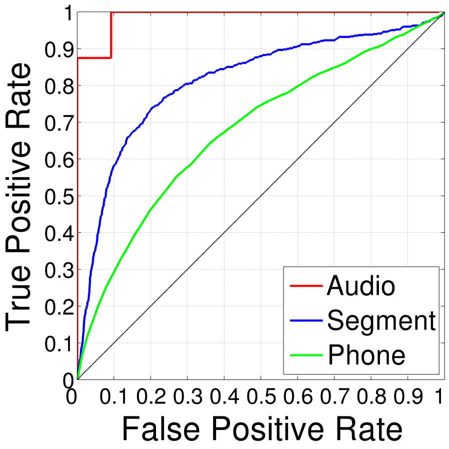
Tab. 4.3: Pearson correlation for audio, segment and phone levels.

4.2 Results on Held-out Data

Using the best scoring system proposed in the previous section, a final evaluation was carried out on "fresh" data in the held-out set. Note that none of the speakers in this set were present in the development set. First, binary classification results are presented and then the outcome considering all classes.

4.2.1 Binary Classification Results

Figure 4.4 and Table 4.5 show the ROC curve and the area under such curve for the held-out set. When compared with the results for the development set (Scheme 1 at Figure 4.6 and Table 4.2), the held-out results are comparable and even slightly better for audio and segment levels. These findings on fresh data show that the decisions made during the development process were general enough to guarantee good results on new data.



Tab. 4.4: ROCs for audio, segment and phone levels.

Audio	Segment	Phone
0.982	0.813	0.743

Tab. 4.5: Area under the ROC curve at audio, segment and phone levels.

4.2.2 Multiclass Classification Results

The normalized histograms for each level are in Figures 4.18, 4.19 and 4.20. Less smooth shapes are seen here due to a lower amount of data in comparison with the development set (Figures 4.11, 4.12 and 4.13). Similar results are seen in this case with classes corresponding to better pronunciation on the left and worse pronunciation classes on the right.

In reference to the dispersion, the same behavior for worse pronunciation classes having "flatter" curves is seen at phone and segment level. However, the number of instances at audio level is so low that it is difficult to see the same compact or disperse patterns.

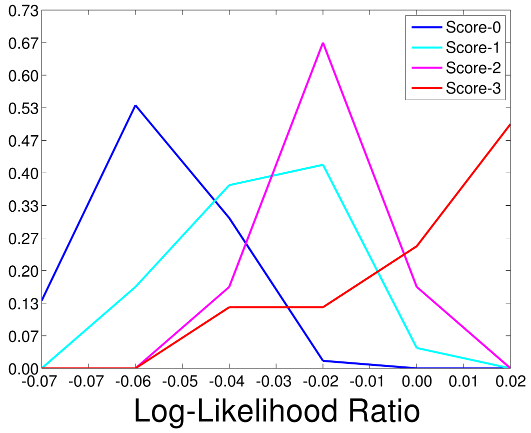


Fig. 4.18: Histogram of all classes for audio level.

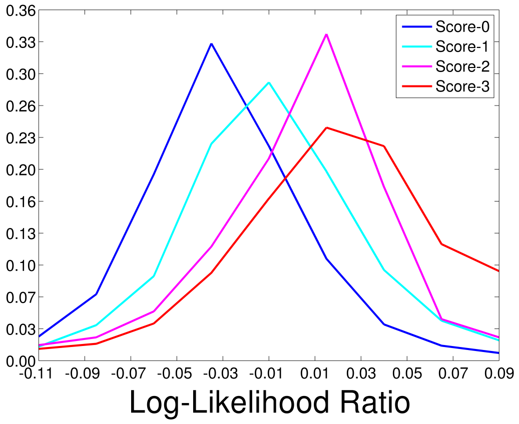


Fig. 4.19: Histogram of all classes for segment level.

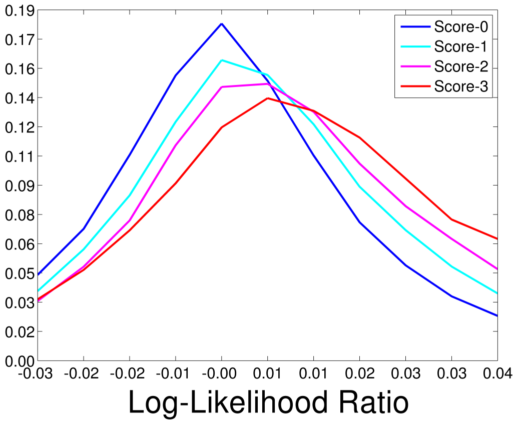


Fig. 4.20: Histogram of all classes for phone level.

Following the analysis about the correlation between the scores and labels done for the development set, the analogous results are presented in Figures 4.21, 4.22, 4.23 and 4.24.

It is possible to see the same behavior for all levels. Also for the held-out set the Score-3 class had more variance than the others showing the same behavior that in the development set and conforming with previous observations in the normalized histograms.

Finally, Table 4.6 has the values for Pearson correlation for the held-out set which, in comparison with the previous reported values (Table 4.3) are very similar.

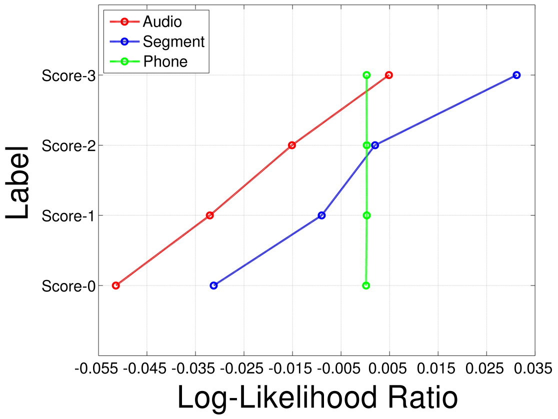


Fig. 4.21: Mean of instances for each class for each level.

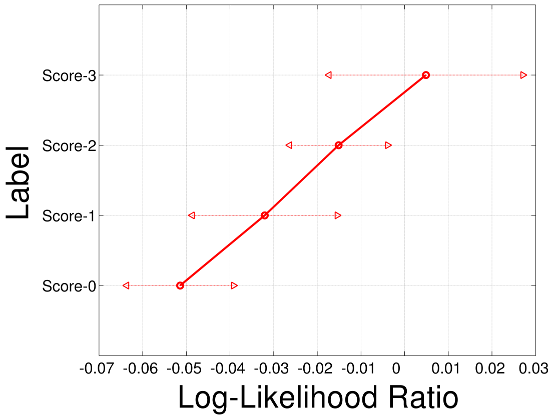


Fig. 4.22: Mean and mean ± standard deviation for audio level.

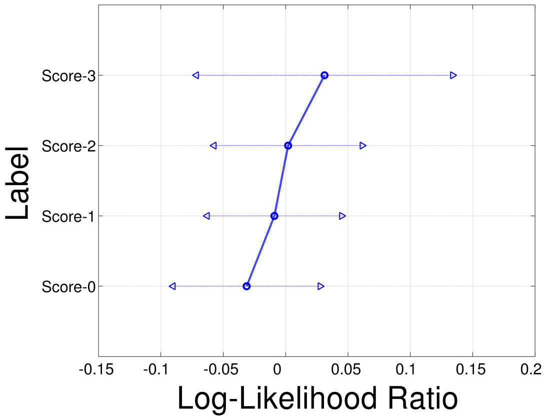


Fig. 4.23: Mean and mean ± standard deviation for segment level.

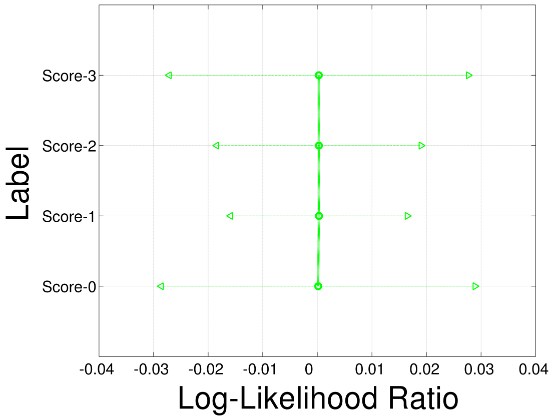


Fig. 4.24: Mean and mean ± standard deviation for phone level.

Audio	Segment	Phone
0.773	0.284	0.002

Tab. 4.6: Pearson correlation for audio, segment and phone levels.

4.3 Comparison with previous results

In this section a comparison of the results obtained in this work with those discussed in 1.2.2 is carried out.

Given the bad performance seen for the LLK method and in order to compare our results with those in previous works, the correlation was computed between the pronunciation scores and the speaker classes at audio level. The correlation for $\tau = 0$ was 0.05. This value for the adaptation parameter was selected for showing one of the best performances for LLR at audio level. A τ value of 10 could have also been selected since they are similar.

The correlation reported in [36], [13] and [29] for the LLK method was between 0.4 and 0.5 in all cases, a much better performance than the 0.05 observed for our setup. Although the specific parameters of the ASR model were not available in those articles, the models used to obtain those results and the one here presented followed the same approach. The main difference in the model itself was that no adaptation was used in those works and a native model was trained with a large amount of native speakers' data. Nevertheless, the amount of data used to adapt the models in this work (around one and a half hours of speech) should be enough to obtain a model adapted to the new domain. Thus this difference is not relevant.

However, another essential difference in the setups are the corpora used in each case. The corpus used in those three articles had two main differences with the one used here. On one hand, the speech quality was different. For this work we used spontaneous telephone conversations in which background noise or signal distortion were common. On the other hand, the previous works ([36], [13] and [29]) used the same corpus, recorded in quiet offices using a high-quality microphone as described in [36]. Secondly, that corpus consisted in mostly read speech for natives and only read speech for non-natives. In comparison with the Fisher corpus, in which all speech are conversations, there is also a great mismatch in the type of speech.

These two differences have a great impact in the recognition quality of an ASR system in general. Worse quality recordings are more difficult to recognize and spontaneous speech is more challenging than read speech for an ASR system. The LLK value will then be a consequence of both the pronunciation quality and the signal's characteristics: a low LLK could signal either a badly pronounced utterance or a mismatch between the utterance and the train conditions.

On the contrary, LLR is the difference between two log-likelihoods, one corresponding to the native model and the other to the non-native model. Both values are influenced by the same aspects described for LLK; however, both of them vary in the same manner when the signal has peculiarities such as noise. When subtracting the two likelihoods, the effect of the acoustic conditions in the two log-likelihoods may compensate each other, therefore reducing the influence of signal distortions. Although this is a reasonable hypothesis to explain the observed results, an experiment with controlled "clean" and "noisy" data from the same set of speakers should be conducted in order to confirm it.

Since the LLR method was used for mispronunciation detection but not for pronunciation scoring, the published results used for comparison use different methods. An aspect relevant for the pronunciation scoring task are the characteristics of the corpora used. In this work, recorded telephone conversations were used to train and test the system. This type of speech has a certain degree of degradation in the quality because of background noise and channel distortion. Furthermore, conversations are usually different in pace and clarity of language than read speech.

For those works in which the quality of data was comparable to the quality of the corpora used for this work ([5] and [23]), the results are similar in terms of correlation at audio level. Cucchiarinni et al. ([5]) reported a correlation of 0.75 at speaker level[2] with 1 minute of telephone read sentences and 0.5 at speaker level with around 1 minute of spontaneous speech in a classroom. Hönig et al. ([23]) reported a speaker level correlation of 0.57 for 10 minutes of headset-recorded speech in noisy environments. Our results, with 0.77 of correlation in the held-out set, are similar to those obtained by Cucchiarinni et al. However, they used only one minute of telephone read speech. Our method, with only one minute of spontaneous conversational speech obtained around 0.3 of correlation showing that more data is needed with our method to obtain comparable results. Nevertheless, the results obtained by Hönig et al. are worse than ours and with the double of data per speaker.

The level of correlation at segment level[3] obtained by our method in the held-out set is only 0.28. The correlation at sentence level was only reported for works with clean data ([36], [13] and [45]) and was 0.58 for the best result. One possible reason for this is that the labels in our

[2] Comparable to the audio level in this work.

[3] Segment level can be compared with the sentence level reported in other works for being of similar durations.

work were inherited from the audio level labels. With a segment level rating, the correlation at this label could possibly improve. Furthermore, the corpus used in that work consisted mostly in read speech recorded in quiet offices with high-quality microphones, a condition very different from the corpus used for this work.

5. CONCLUSIONS

In this work we presented an automatic pronunciation scoring system based on two different methods that we called LLK and LLR. They were based on log-likelihoods and log-likelihood ratios of models trained with different populations of speakers. The first one did not produce acceptable results as reported in previous works using acoustically clean corpora and read speech. One possible reason is that in this work the corpus was noisier (channel distortion and background noise) and conversational. The log-likelihood ratio was obtained using a native model and a non-native model obtained by adaptation to non-native speakers. The proposed LLR method was based on two previously approaches for the related task of mispronunciation detection. Our method was applied to the pronunciation scoring task and the results here presented are comparable to those in the literature for other systems on the same task.

Different configurations to obtain the native and non-native models were evaluated. All of them were based on domain adaptation from a previous model trained with a large native corpus. Maximum a posteriori adaptation was used and different levels of adaptation were evaluated. The best configuration was obtained with a scheme in which the native model was obtained adapting the initial model and the non-native model was obtained by adapting this new native model to the non-native data. This model produced higher gains in performance when less data was available for adaptation in comparison with a scheme in which both native and non-native models were created by adapting directly to the initial model. Furthermore, similar results to those obtained using all data, around 300 seconds per speaker per conversation, were observed using only 200 seconds per speaker per conversation.

Different levels of adaptation were better for different regions of speech. High levels such as complete conversations benefit more from a higher degree of belief on the adaptation data while lower region levels such as phones benefit more with greater belief on previous data. This is due to the fact that lower level regions are more sensitive to noise and thus the adaptation must be done more cautiously. On the other hand, longer regions have more information to average over the potential noise in the model, and thus a stronger adaptation is better since it allows for more detailed modeling of pronunciation details in each population.

Different approaches to obtain phone-level log-likelihoods were evaluated and the average of the log-likelihoods at frame level had the best results compared to the sum and the average multiplied by the logarithm of the duration of the phone. This suggests that the duration of the phones should not be taken into account when computing the log-likelihoods and all phones should be treated independently of their duration.

An heuristic which used the length of the recognized phones was devised in order to cope with forced alignment errors. This heuristic did not produce considerable improvements showing that the phones with extremely long durations due to alignment errors did not influence the scores in a high degree.

On the whole, the pronunciation scoring system presented a good level of correlation at audio level comparable to the results obtained in previous works using corpora with similar characteristics. This shows that the LLR method is capable of obtaining good results even in adverse conditions, similar to those expected in a school or house when using a CALL system with a headset.

Future Work

With the experiments proposed in this work, some questions have arisen that need further investigation. These inquiries lead to possible lines of research as described next:

- Further investigation needs to be done with controlled “clean” and “noisy” data to understand how robust is the LLR method to noise in the signal.
- Evaluate the system with a broader set of speakers regarding the pronunciation quality, i.e. also including beginner learners.
- Evaluate the system with “noisy” but read data (instead of conversations) since that task is more similar to what is expected in a CALL system.
- Evaluate the results when using a more complex ASR model as initial model. A possible approach for this could be to implement the system using a DNN-based model.

BIBLIOGRAPHY

[1] Jared Bernstein, Michael Cohen, Hy Murveit, Dimitry Rtischev, and Mitchel Weintraub. “Automatic evaluation and training in English pronunciation.” In: *ICSLP*. Vol. 90. 1990, pp. 1185–1188.

[2] Jared Bernstein and Horacio Franco. “Speech recognition by computer”. In: *Principles of experimental phonetics. St. Louis: Mosby* (1996).

[3] Tobias Cincarek, Rainer Gruhn, Christian Hacker, Elmar Nöth, and Satoshi Nakamura. “Automatic pronunciation scoring of words and sentences independent from the non-native’s first language”. In: *Computer Speech & Language* 23.1 (2009), pp. 65–88.

[4] Stephen Cox and Srinandan Dasmahapatra. “High-level approaches to confidence estimation in speech recognition”. In: *IEEE Transactions on Speech and Audio processing* 10.7 (2002), pp. 460–471.

[5] Catia Cucchiarini, Helmer Strik, Diana Binnenpoorte, and Lou Boves. “Pronunciation evaluation in read and spontaneous speech: A comparison between human ratings and automatic scores”. In: *Proceedings of the New Sounds* (2000).

[6] Catia Cucchiarini, Helmer Strik, and Lou Boves. “Different aspects of expert pronunciation quality ratings and their relation to scores produced by speech recognition algorithms”. In: *Speech Communication* 30.2 (2000), pp. 109–119.

[7] Catia Cucchiarini, Helmer Strik, and Lou Boves. “Quantitative assessment of second language learners’ fluency by means of automatic speech recognition technology”. In: *The Journal of the Acoustical Society of America* 107.2 (2000), pp. 989–999.

[8] Steven Davis and Paul Mermelstein. “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences”. In: *IEEE transactions on acoustics, speech, and signal processing* 28.4 (1980), pp. 357–366.

[9] Joost van Doremalen, Catia Cucchiarini, and Helmer Strik. “Automatic detection of vowel pronunciation errors using multiple information sources”. In: *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on.* IEEE. 2009, pp. 580–585.

[10] Farzad Ehsani and Eva Knodt. “Speech technology in computer-aided language learning: Strengths and limitations of a new CALL paradigm”. In: *Language Learning & Technology* 2.1 (1998), pp. 45–60.

[11] Horacio Franco, Luciana Ferrer, and Harry Bratt. “Adaptive and discriminative modeling for improved mispronunciation detection”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE. 2014, pp. 7709–7713.

[12] Horacio Franco and Leonardo Neumeyer. “Calibration of machine scores for pronunciation grading.” In: *ICSLP*. 1998.

[13] Horacio Franco, Leonardo Neumeyer, Yoon Kim, and Orith Ronen. “Automatic pronunciation scoring for language instruction”. In: *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on.* Vol. 2. IEEE. 1997, pp. 1471–1474.

[14] Horacio Franco, Leonardo Neumeyer, María Ramos, and Harry Bratt. “Automatic detection of phone-level mispronunciation for language learning.” In: *EUROSPEECH*. 1999.

[15] Mark Gales and Steve Young. “The application of hidden Markov models in speech recognition”. In: *Foundations and trends in signal processing* 1.3 (2008), pp. 195–304.

[16] J-L Gauvain and Chin-Hui Lee. “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains”. In: *IEEE transactions on speech and audio processing* 2.2 (1994), pp. 291–298.

[17] Hiroshi Hamada and Ryohei NAKATSU. “Automatic evaluation of English pronunciation based on speech recognition techniques”. In: *IEICE TRANSACTIONS on Information and Systems* 76.3 (1993), pp. 352–359.

[18] Alissa M Harrison, Wai-Kit Lo, Xiaojun Qian, and Helen Meng. “Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training.” In: *SLaTE*. 2009, pp. 45–48.

[19] Hynek Hermansky. “Perceptual linear predictive (PLP) analysis of speech”. In: *the Journal of the Acoustical Society of America* 87.4 (1990), pp. 1738–1752.

[20] Daniel Herron, Wolfgang Menzel, Eric Atwell, Roberto Bisiani, Fabio Daneluzzi, Rachel Morton, and Juergen A Schmidt. “Automatic localization and diagnosis of pronunciation errors for second-language learners of English”. In: *Sixth European Conference on Speech Communication and Technology, EUROSPEECH 1999, Budapest, Hungary, September 5-9, 1999*. ISCA. 1999.

[21] Steven Hiller, Edmund Rooney, John Laver, and Mervyn Jack. “SPELL: An automated system for computer-aided pronunciation teaching”. In: *Speech Communication* 13.3-4 (1993), pp. 463–473.

[22] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97.

[23] Florian Hönig, Anton Batliner, and Elmar Nöth. “Automatic assessment of non-native prosody annotation, modelling and evaluation”. In: *International Symposium on Automatic Detection of Errors in Pronunciation Training (IS ADEPT)*. 2012, pp. 21–30.

[24] Florian Hönig, Anton Batliner, Karl Weilhammer, and Elmar Nöth. “Islands of failure: employing word accent information for pronunciation quality assessment of English L2 learners.” In: *SLaTE*. Citeseer. 2009, pp. 41–44.

[25] Kazunori Imoto, Yasushi Tsubota, Tatsuya Kawahara, and Masatake Dantsuji. “Modeling and automatic detection of English sentence stress for computer-assisted English prosody learning system”. In: *Acoustical science and technology* 24.3 (2003), pp. 159–160.

[26] Biing-Hwang Juang and Lawrence R Rabiner. “Automatic speech recognition–a brief history of the technology development”. In: *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara* 1 (2005), p. 67.

[27] Sandra Kanters, Catia Cucchiarini, and Helmer Strik. “The goodness of pronunciation algorithm: a detailed performance study.” In: *SLaTE* 2009 (2009), pp. 2–5.

[28] Hiroshi Kibishi and Seiichi Nakagawa. “New Feature Parameters for Pronunciation Evaluation in English Presentations at International Conferences.” In: *INTERSPEECH*. 2011, pp. 1149–1152.

[29] Yoon Kim, Horacio Franco, and Leonardo Neumeyer. “Automatic pronunciation scoring of specific phone segments for language instruction.” In: *Eurospeech*. 1997.

[30] David A van Leeuwen and Joost van Doremalen. “Calibration of Phone Likelihoods in Automatic Speech Recognition”. In: *arXiv preprint arXiv:1606.04317* (2016).

[31] Christopher J Leggetter and Philip C Woodland. "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models". In: *Computer Speech & Language* 9.2 (1995), pp. 171–185.

[32] Jüri Lember, Alexey Koloydenko, et al. "The adjusted Viterbi training for hidden Markov models". In: *Bernoulli* 14.1 (2008), pp. 180–206.

[33] Ben Milner. "A comparison of front-end configurations for robust speech recognition". In: *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on.* Vol. 1. IEEE. 2002, pp. I–797.

[34] N Moustroufas and Vassilios Digalakis. "Automatic pronunciation evaluation of foreign speakers using unknown text". In: *Computer Speech & Language* 21.1 (2007), pp. 219–230.

[35] Leonardo Neumeyer, Horacio Franco, Vassilios Digalakis, and Mitchel Weintraub. "Automatic scoring of pronunciation quality". In: *Speech communication* 30.2 (2000), pp. 83–93.

[36] Leonardo Neumeyer, Horacio Franco, Mitchel Weintraub, and Patti Price. "Automatic text-independent pronunciation scoring of foreign language student speech". In: *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on.* Vol. 3. IEEE. 1996, pp. 1457–1460.

[37] Daniel Povey. *Kaldi ASR.* URL: `http://kaldi-asr.org/`.

[38] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. "The Kaldi speech recognition toolkit". In: *IEEE 2011 workshop on automatic speech recognition and understanding.* EPFL-CONF-192584. IEEE Signal Processing Society. 2011.

[39] Josef Psutka, Ludek Müller, and Josef V Psutka. "Comparison of MFCC and PLP parameterizations in the speaker independent continuous speech recognition task." In: *INTERSPEECH.* 2001, pp. 1813–1816.

[40] Xiaojun Qian, Frank K Soong, and Helen M Meng. "Discriminative acoustic model for improving mispronunciation detection and diagnosis in computer-aided pronunciation training (CAPT)." In: *INTERSPEECH.* 2010, pp. 757–760.

[41] Lawrence R Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[42] Lawrence R Rabiner and Ronald W Schafer. "Introduction to digital speech processing". In: *Foundations and trends in signal processing* 1.1 (2007), pp. 1–194.

[43] Sean Robertson, Cosmin Munteanu, and Gerald Penn. "Pronunciation Error Detection for New Language Learners". In: *Interspeech 2016* (2016), pp. 2691–2695.

[44] Catherine L Rogers, Jonathan M Dalby, and Gladys DeVane. "Intelligibility training for foreign-accented speech: A preliminary study". In: *The Journal of the Acoustical Society of America* 96.5 (1994), pp. 3348–3348.

[45] Orith Ronen, Leonardo Neumeyer, and Horacio Franco. "Automatic detection of mispronunciation for language instruction." In: *EUROSPEECH.* 1997.

[46] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks". In: *Proceedings of the conference on empirical methods in natural language processing.* Association for Computational Linguistics. 2008, pp. 254–263.

[47] Helmer Strik, Khiet P Truong, Febe De Wet, and Catia Cucchiarini. “Comparing classifiers for pronunciation error detection.” In: *Interspeech*. 2007, pp. 1837–1840.

[48] Carlos Teixeira, Horacio Franco, Elizabeth Shriberg, Kristin Precoda, and M Kemal Sönmez. “Prosodic features for automatic text-independent evaluation of degree of nativeness for language learners.” In: *INTERSPEECH*. 2000, pp. 187–190.

[49] Khiet Truong, Ambra Neri, Catia Cucchiarini, and Helmer Strik. “Automatic pronunciation error detection: an acoustic-phonetic approach”. In: *InSTIL/ICALL Symposium 2004*. 2004.

[50] Yow-Bang Wang and Lin-Shan Lee. “Improved approaches of modeling and detecting error patterns with empirical analysis for computer-aided pronunciation training”. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2012, pp. 5049–5052.

[51] Si Wei, Guoping Hu, Yu Hu, and Ren-Hua Wang. “A new method for mispronunciation detection using support vector machine based on pronunciation space models”. In: *Speech Communication* 51.10 (2009), pp. 896–905.

[52] Silke M Witt. “Automatic error detection in pronunciation training: Where we are and where we need to go”. In: *Proc. IS ADEPT* 6 (2012).

[53] Silke M Witt and Steve J Young. “Phone-level pronunciation scoring and assessment for interactive language learning”. In: *Speech communication* 30.2 (2000), pp. 95–108.

[54] Su-Youn Yoon, Mark Hasegawa-Johnson, and Richard Sproat. “Automated pronunciation scoring using confidence scoring and landmark-based SVM.” In: *Interspeech*. 2009, pp. 1903–1906.

[55] Su-Youn Yoon, Mark Hasegawa-Johnson, and Richard Sproat. “Landmark-based automated pronunciation error detection.” In: *Interspeech*. 2010, pp. 614–617.

[56] Steve J Young, Julian J Odell, and Philip C Woodland. “Tree-based state tying for high accuracy acoustic modelling”. In: *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics. 1994, pp. 307–312.

[57] Feng Zhang, Chao Huang, Frank K Soong, Min Chu, and Renhua Wang. “Automatic mispronunciation detection for Mandarin”. In: *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2008, pp. 5077–5080.