Dialect and Accent Recognition

Hamid Behravan

Master's Thesis

UNIVERSITY OF
EASTERN FINLAND

University of Eastern Finland

School of Computing

December, 2012

Abstract

State of the art speech recognition systems are concentrating on modeling the speech variances among different speakers such as dialects or accents of the spoken language, the speaker gender, as well as some physiological variances such as differences in vocal tract length. These variations lead to difficulties in modeling large-scale speaker-independent systems. This thesis focuses on recognizing the dialect or accent of a given speech utterance, and demonstrates how parameter tuning affects the system performance.

In this thesis, we construct a baseline dialect recognition system based on frame-based spectral modeling techniques and describe how we can improve the system performance by tuning the baseline system parameters. Then the performance of the baseline system is compared with identity vectors (i-vectors) based dialect recognition system. The main contribution of this study is to observe the sensitivity of the evaluation metrics on parameters of the designed dialect and accent recognition system.

Our experiments are based on Callfriend corpus, and from this corpus English, Mandarin and Spanish languages were selected. For each experiment, three different evaluation metrics are used, identification error rates, equal error rates and the detection cost function.

The best results achieved in i-vector based dialect recognition system for all of three selected languages. The English language achieved its best results at minimum detection cost function of 0.0807, identification error rate of 28.43% and equal error rate of 11.31%, Mandarin language achieved its best results at minimum detection cost function of 0.0543, identification error rate of 23.56% and equal error rate of 8.41%, and finally Spanish language achieved its best result at minimum detection cost function of 0.0569, identification error rate of 28.24% and equal error rate of 9.12%. The results indicate that Mandarin dialects have the most distinctive characteristics compared to other two languages. Moreover, Regarding the sensitivity of the results to tuning parameters, the recognition system is more sensitive to changes in number of GMM components and VAD thresholds than other tuning parameters. VTLN, RASTA filtering, addition of mel-cepstral coefficients to SDC feature vectors, and feature normalization appear in next orders, respectively.

keywords: Dialect and Accent Recognition, Spectral Modeling, Phonotactic modeling, Shifted Delta Cepstral Coefficients, Universal background Modeling, Vocal Tract Length Normalization, Identity Vectors.

Contents

List of Figures:

List of Tables:

Abbreviations:

OED: Oxford English Dictionary

VTLN: Vocal tract length normalization

I-vector: Identity vector

MMI: Maximum mutual information

SDC: Shifted delta cepstral coefficients

RP: Received pronunciation

PMVDR: perceptual minimum variance distortion-less response

PR: Puerto Rico

LRE: Language Recognition Evaluation

KL: Kullback-Leibler

VAD: Voice activity detection

LDA: Linear discriminant analysis

EM: Expectation maximization

minDCF: Minimum Value of the Detection Cost Function

IDerror: Identification error rate

EER: Equal error rate

RASTA: Relative spectral filtering

IIR: Infinite impulse response

MFCC: Mel-frequency cepstral coefficients

UBM: Universal background modeling

VTLN: Vocal tract length normalization

CMV: Cepstral mean variance

Acknowledgement

I would like to thank Ville Hautamäki and Tomi Kinnunen, my kind supervisors, for providing me with constant helpful support and feedbacks. They made me interested in the topic and taught me how to do research.

My sincere thanks to my best friends who have been always there to encourage me and make me smile in all the difficulties I faced in my life. I am deeply happy and thankful to have such friends.

Last but not least, my sincere gratitude goes to my mother and father for their all kindness and encouragement during my studies, without them this thesis would have been very hard to finish.

تقدیم به پدر و مادر مهربانم

و

تمام مردم خوب سرزمینم

Chapter 1

Introduction

1.1 Background

Dialect/accent refers to different ways of pronouncing/speaking a language within a community. Examples could be American English vs. British English speakers or the Spanish speakers in Spain vs. Caribbean.

During the past few years, there have been significant attempt to automatically recognize the dialect or accent of a speaker given his or her speech utterance [5, 7, 9, 30]. Recognition of dialects or accents of speakers prior to automatic speech recognition (ASR) helps in improving performance of the ASR systems by adapting the ASR acoustic and/or language models appropriately [44]. Moreover, in applications such as telephone-based assistant systems, by recognizing the dialect or accent of the caller and then connecting the caller to agent with similar dialect or accent will produce more user friendly environment for the users of the application [45].

This thesis provides the following contributions. First, a baseline dialect and accent recognition system is designed so that new ideas could be developed on it. Second, the system recognition sensitivity to front-end and back-end parameters are experimented. Regarding the designed baseline system, a new modeling approach, identity vectors (i-vectors), is coupled with baseline system so that the system get closer to the state of the art dialect and accent recognition systems.

1.2 Research problem, objectives and questions

The main objectives of this research are to find the effect of parameters of the base-line dialect and accent recognition system, observing the effects of the a new modeling techniques added to the baseline system, and finally compare the recognition performance of selected dialects. This comparison shows that which language dialects have the most distinctive characteristics in their phonemes, and other dialect or accent dependent attributes.

The study attempts to find answers to the following research questions:

- What are the optimal tuning parameters for the base-line dialect and accent identification system?
- How does the overall system performance change when VTLN and i-vectors are added to the base-line system?
- Does the identification performance significantly vary among the selected languages? If yes, which of the languages give the highest and lowest recognition accuracies?

1.3 Thesis Structure

In chapter 2, we will look at the automatic dialect and accent recognition systems by reviewing what has already been done in literature and describing how the baseline system is developed and what components have been used to construct it. Then in Chapter 3 which is the most important chapter of this thesis, we will present our different experiments we have done in this study together with analysis of these experiments. Finally, in Chapter 4 conclusions and future works are explained.

Chapter 2

Dialect and Accent Recognition

In this chapter, we will look at the theory behind the dialect and accent recognition systems. The first part of this chapter, we describe the differences between dialects, accents and the styles of speakers, how the linguistics differentiate dialects as well as reviewing different approaches in automatic dialect and accent recognition systems. Finally, in the end of this chapter, we discuss which approach has been used in our research and will justify the reason behind this selection.

2.1 Dialects vs. Accents vs. Styles

In this section, we describe three different linguistic variations that appear in any language. Two of these categories are specified by regional variations as in pronunciations (*accents*), word selection and grammar (*dialects*), and by sociological variations as in different speaking styles due to age, situation and gender. Knowing all of these variables creates insight into social, historical and geographical factors of language being used in the society [9].

2.1.1 Dialects

Dialects are varieties of speech within a specified language. The Oxford English Dictionary (OED) describes dialects as *"one of the subordinate forms or varieties of a language arising from local peculiarities of vocabulary, pronunciation and idiom"*. These variations can exist at all linguistic levels, i.e. vocabularies, idioms, grammars and pronunciation. Some examples in case of South and North English dialects are

- South: "Howdy"; North: "Hello"

- South: "Fixin to"; North: "About to"

and in case of Canary Island and Madrid Spanish dialects are [1]

- Madrid: pronunciation of c and z like the sound "th", as in *Los centros* sounds like "Los thentros".
- Canary: pronunciation of c and z like the sound "s", as in *Centros* sounds like "sentros".

Dialects of the specific language differ from each other, but they are still understandable to the speakers of another dialect of the same language. Differences among dialects are mainly due to regional and social factors and these differences vary in terms of pronunciation, vocabulary, and grammar [3]. For example, the sentence "she were wearing a sunglass" might sound unusual, but in some dialects in northern England and the Midlands, many speakers use the past tense of "to be" by saying "I were, you were, he, she and it were, we were and they were". This means that the verb is unchanged for person, while speakers of Standard English use "I was and he, she and it was". This example indicates how standard grammars of a language might be influenced by regional dialect differences. On the other hand, social factor shows that members of a specific socioeconomic class such as working-class dialects, might have different dialects compared to high-class businessman. So the way a person speaks his/her language is highly influenced by both his/her social status and his/her region of origin.

2.1.2 Accents

Accents are defined as varieties in pronunciations of a certain language and refers to the sounds that exists in a person's language. Therefore, everybody has an accent. Generally, accents differ in two subjects, *phonetic* and *phonological* [10,]. When accents differ in phonetic, there are same set of phonemes in both accents, but some of these phonemes are realized differently. For example, the phoneme *'e'* in *dress* is pronounced as 'ɜ' in England, and *'e'* in Wales. Another example, the phoneme *'u'* in *strut* is pronounced as 'ʌ' in England, and 'ʊ' in Wales. Differences in stress and intonation are also refer to *phonetic* category.

On the other hand, *phonological* refers to those accents which have different number of phonemes from another and often the identity of phonemes are also different. Examples are *made* or *waste* which are pronounced as *'e'*ɪ in England and as *'e:'* in Wales [10].

What is clear in accent description is that unlike the dialects, accents only cover a small group of variations which could occur in a certain language.

2.1.3 Styles

Another type of linguistic variation in a certain language is caused due to different styles of speakers within that language. Styles generally refer to the mood of the speaker and the situations in which the speaker is placed. This factor differs from dialect and accent variations so that dialect and accent is the way a certain language is spoken among many people of a society, whereas styles refer to the spoken language of the same person in different situations.

As mentioned, styles of a speaker depend on situational factors such as

- who is he/she speaking with
- what is the spoken topic about
- where is the conversation taking place

In all of these situations, one speaker might select a different tone of voice in his voice. For example, in *careful styles*, more attention is paid to speech, whereas in *casual styles*, there will be less attention on the monitoring of speech [10].

2.2 Approaches in Dialect and Accent Recognition Systems

How do the the automatic dialect and accent recognizer systems work? In general, these system uses two different approaches, *phonotactic* and *spectral* approaches. In the following section, I will briefly review the core idea behind these approaches and will explain that which of these techniques has been used in this study and for what reason the approach is selected.

2.2.1 Phonotactic Approaches

The phonotactic approach in dialect and accent recognition recognition is based on the hypothesis that dialects or accents differ in their *phone sequence distributions*. In other words, texts of a same language can be recognized by these character distributions. Using the probabilistic frameworks such as the ones mentioned in [15] and assuming that the phonemes prior distributions are uniform, the dialect or accent recognition problem can be written as

$$\arg\max_i P(Q|D_i)$$

$P(Q|D_i)$ denotes the conditional probability of the Q given D_i, where Q represents the sequence of phonemes and D_i represents the target dialect i.

As explained above, dialect differences are mostly due to regional and social factors, and these differences are seen in vocabulary, grammar and pronunciation. In fact, phone sequence recognizers capture these subtle differences within the dialects of a certain language.

One of the methods which uses the phonotactic approaches is PRLM (Phone Recognition followed by Language Modeling) [14]. In this method, for dialect or accent recognition, the phones of the training speech utterances of a specific dialect or accent are first recognized using a single phone recognizer. Then an N-gram model, η_i, [1] is trained on the detected phone sequences.

During the recognition process, The phone sequences ,Q, are extracted from the given test utterance, and then the likelihood of each phone sequence is computed given the trained N-gram dialect models. As an example, if $N = 3$, then the likelihood is computed as [14]

$$P(Q = q_q, q_2, \ldots, q_k|\eta_i) = P(q_1|\eta_i)P(q_2|\eta_i)\prod_{j=3}^{k} P(q_j|q_{j-1}, q_{j-2}, \eta_i)$$

The dialect with the N-gram model that gives the maximum likelihood is selected as the hypothesized dialect of the given speech utterance.

Phonotactic-based approaches have been the baseline system of some new conducted research in dialect and accent recognition area. For example, [12] mentioned that, due to small differences in phonemes of different dialect or accents of a specific language, it is essential to choose a small subset of discriminative features that can be robustly estimated. At the same time non-discriminative and noisy features can be removed. This feature subset selection led to more than 20% relative improvement in accent recognition rate irrespective to the choice of classifier. In another work, [4] introduces an approach to dialect recognition task which is based on context-dependent (CD) phonetic differences between dialects as well as phonotactics. This approach demonstrates which phones in what contexts considerably distinguish between dialect pairs. The method tested on four spontaneous telephone speech Arabic dialects with 30s length utterances [46]. Since the number of available speakers were different in each of these four dialects, the equal number of utterances were used for testing procedure. The Results indicated that this approach performs considerably better than the GMM-UBM system and also PRLM system at 5% absolute equal error rate (EER).

[1] an N-gram is an adjacent sequence of n items from a given sequence of text or speech. An N-gram could be any combination of letters, phonemes or syllables.

2.2.2 Spectral Approaches

The spectral modeling approach for dialect and accent recognition is based on the hypothesis that dialects or accents discriminates in terms of their spectral (acoustic) features. In this approach, speech utterances are represented by a set of spectral feature vectors and the recognition is based on maximum likelihood estimation.

Most spectral-based dialect and accent recognition systems apply a gaussian mixture model (GMM) [49] to model the spectral distribution of each dialect or accent. In this approach, the spectral vectors are assumed to be statistically independent and the feature space is represented by mean vectors, μ, the covariance matrix, C and the mixture weights, w.

During recognition, given an utterance spectral feature vector, a model will be selected so that gives the maximum likelihood according to the following equation

$$\arg\max_{i} \prod_{t=1}^{T} P(a_t|D_i)$$

where a_t is the feature vector at frame t, D_i is the target dialect i, and T is the total number of frames.

Spectral-based approaches have also been the baseline approaches in many recent dialect and accent recognition systems. [27] has argued that GMM-based language recognition systems are sensitive to feature variability caused by non-language factors, such as speaker and channel distortions. In this work, he proposed a new feature-space transform based on constrained maximum likelihood linear regression (CMLLR) for compensation of these distortion effects. In another work, [11] used a discriminatively trained Gaussian mixture models and feature compensation using eigen-channel decomposition to compensate the channel and speaker distortions. By using these techniques, they were able to increase their system performance by 10% relative improvements in equal error rate.

The advantage of using back-end classifiers on acoustic scores were investigated in [13], where, it was shown that the use of back-end classifier leads to a consistent improvement for GMM-UBM systems. In fact back-end classifiers train discriminative models based on dialect or accent model scores, where correlated scores are clustered in the same category. So that during recognition process, given a new test utterance model score, a model will be selected so that gives the maximum likelihood. Conceptually, back-end classifiers categorized the correlated scores in the same clusters.

A new set of spectral features are introduced in [8], where perceptual minimum variance distortionless response (PMVDR), were concatenated to

shifted delta cepstral coefficients (SDCs)[2], and the concatenated feature vectors were trained using generalized maximum likelihood estimation (MLE) framework. As most of the dialect or accent discriminative features exist at the upper spectral envelope at the perceptually important harmonics, PMVDR method has the ability to better track the upper spectral envelope of speech spectrum compared to mel-frequency cepstral coefficients (MFCCs) [47]. The key factor in PMVDR computation is that the lterbank is removed and mel-frequency warping directly applied on the fast Fourier transform power spectrum. This novel feature extraction method reported +26.4% relative improvement in dialect recognition rate.

It should be noted that most of the trends in recent spectral approaches in dialect and accent recognition task are towards to channel and speaker distortion compensations. In the coming sections, we will mention the two techniques in which we have used to compensate distortions. These techniques were both applied at feature extraction and dialects modeling processes.

2.3 The modeling approach selected in this Thesis

The approach that we considered in this study is based on spectral modeling techniques. In spectral approaches the dialect identification task is based on the whole utterance including speech and non-speech frames, whereas in phonotactic based approaches the decisions are based on the single phonemes. On the other hand, phonotactic based approaches use Hidden Markov models (HMM) that adds more complication to the dialect and accent recognition task. It should be noted that the current state-of-the-art dialect and accent recognition systems have used phonotactic modeling approaches for years, but in recent years spectral modeling approaches coupled with identity vectors[3] have conquered the best phonotactic systems [21, 40].

[2]SDCs are discussed in Chapter 3.

[3]i-vector systems will be discussed in details in Chapter 3.

Chapter 3

Spectral Dialect and Accent Recognition System

In this chapter, we will discuss the Spectral-based dialect and accent recognition system developed in this study. We will review the different components of the baseline system and we will move on with describing the state of the art dialect and accent recognition system components which we have used in this research.

The system components can be generally divided into front-end and back-end processing parts. In front-end processing section, we will look at those components which are used during feature extraction process, while in back-end processing section, modeling and classification components are described.

3.1 Front-end Processing

3.1.1 Mel-frequency Cepstral Coefficients (MFCCs)

Conventional automatic speech recognition (ASR) systems are constructed in two stages: feature extraction and modeling. Often, the modeling stage is based on hidden Markov models (HMM) or Gaussian mixture models (GMMs) as depicted in Fig. 3.1.

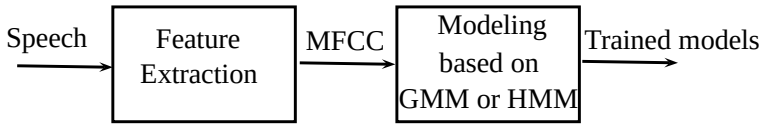

Fig. 3.1: Demonstration of a conventional ASR system.

The feature extraction process is usually a non-invertible (lossy) transformation. Such transformation does not lead to a perfect reconstruction of speech signal, i.e., given only the feature vectors, it is not possible to reconstruct the original speech signal which was used to generate those feature vectors. Fig. 3.2 represents the block diagram for computing MFFCs.

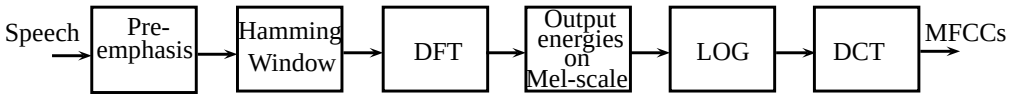

Fig. 3.2: The block diagram of MFCC computation.

Regarding feature extraction process, speech signal is segmented into overlapped frames of length 20 to 30 ms, and then the speech signal is passed into a high-pass filter according to [50]

$$y[n] = x[n] - \beta x[n-1]$$

where $x[n]$ is the input speech signal, $y[n]$ is the pre-emphasized output speech signal and β is an adjustable parameter which is usually between 0.9 and 1. The goal of pre-emphasis is to compensate the high-frequency components of input signal which have been smothered during the sound production mechanism of humans. Moreover, it can also amplify the importance of high-frequency formants in a given speech signal [50].

The next step is to multiply each frame by a windowing function to keep the continuity of the first and the last points in the frame.

$$y^{'}[n] = y[n]w[n]$$

Where $w[n]$ is a windowing function and $y[n]$ is the frame of pre-emphasized speech signal from the previous step. This multiplication has a good effect on eliminating the distortions appeared in the spectral analysis of the signal once a section of the signal (frames) are cut for spectral analysis.

There are various smoothing functions available, but Hamming windowing function [50] is usually used in speech signal processing. Experiments have shown that Hamming windowing has the property to cause less amount of distortions in the spectral analysis of the speech signals [50]. The general form of Hamming window with a control parameter α is given in below equation [50].

$$w[n, \alpha] = (1 - \alpha) - \alpha \cos\left(2\pi n/(N-1)\right),\ 0 \leq n \leq N-1$$

In fact α controls the amount of curvature in the shape of hamming window. In Fig. 3.3 different curves for the hamming window with respect to different values of α is shown.

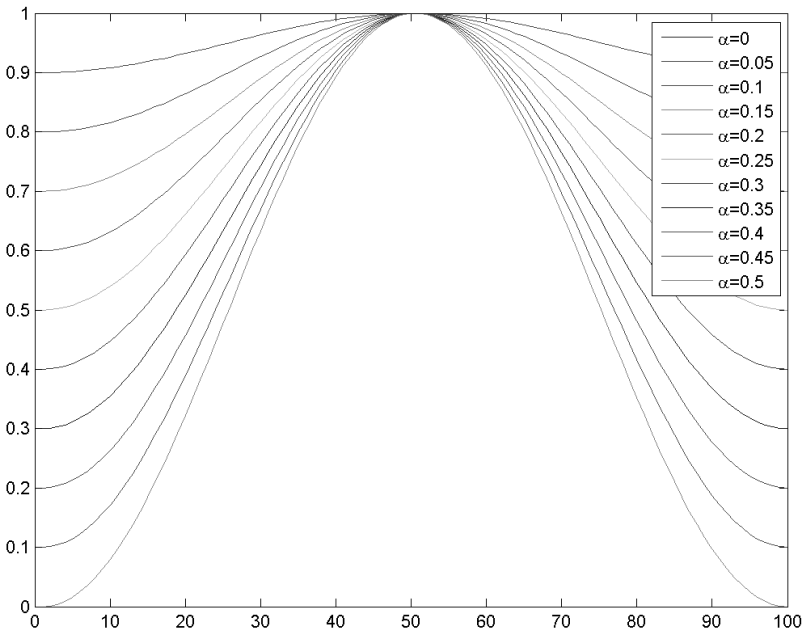

Fig. 3.3: Different curves for the hamming window with respect to different values of α. In this study we used the value of 0.46 for the α.

After windowing, frames are ready for spectral analysis in the MFCC process. Fast Fourier Transform (FFT) is applied on the signal to obtain the magnitude frequency response of each frame.

In the next phase, the magnitude frequency response of FFT is multiplied by a set of N, usually 20, triangular bandpass filters to get the log energy of each triangular bandpass filter. The positions of these filters are equally spaced along the Mel frequency, which is mathematically related to the signal linear frequency f by the following equation [49]

$$mel(f) = 1000 \log_{10}(1 + f/1000)$$

Above equation follows a general form of conversion between Hertz to Mel frequency. Some researchers use different multiplication coefficient as in [41]. Mel-frequency is proportional to the logarithm of the linear frequency, and it generally represents the similar effects in which human's ears mechanism capture the voice signals leading to our aural perception. Fig. 3.4 plots the relationship between the mel and the linear frequencies.

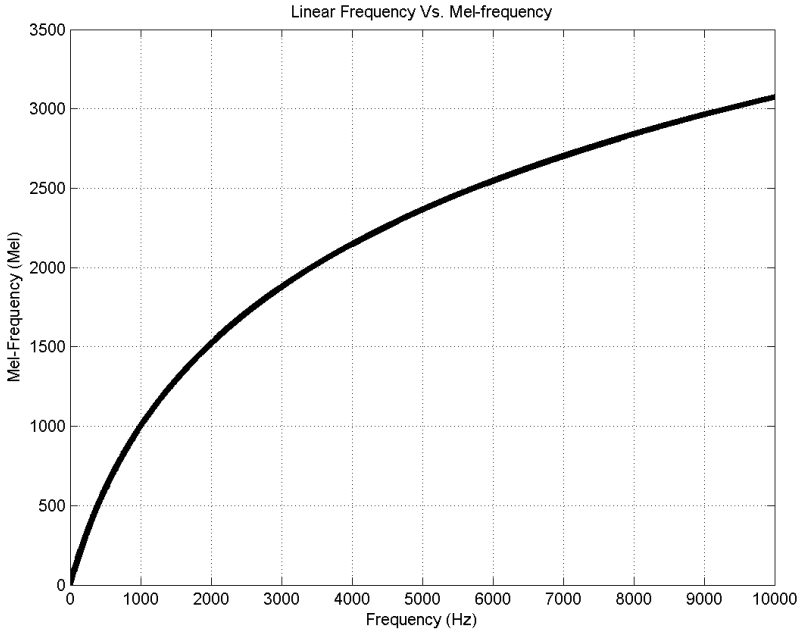

Fig. 3.4: Plot of linear frequency Vs. Mel-frequency.

Practically there are two reasons for using triangular bandpass filters:

1. Smoothing the magnitude spectrum such that the harmonics are flattened in order to obtain the envelope of the spectrum. This indicates that the pitch of a speech signal is generally not presented in MFCCs. As a result, a speech recognition system will behave more or less the same when the input utterances are of the same timbre but with different pitch.

2. Reduce the number of features.

An example of a filterbank is shown in Fig. 3.5.

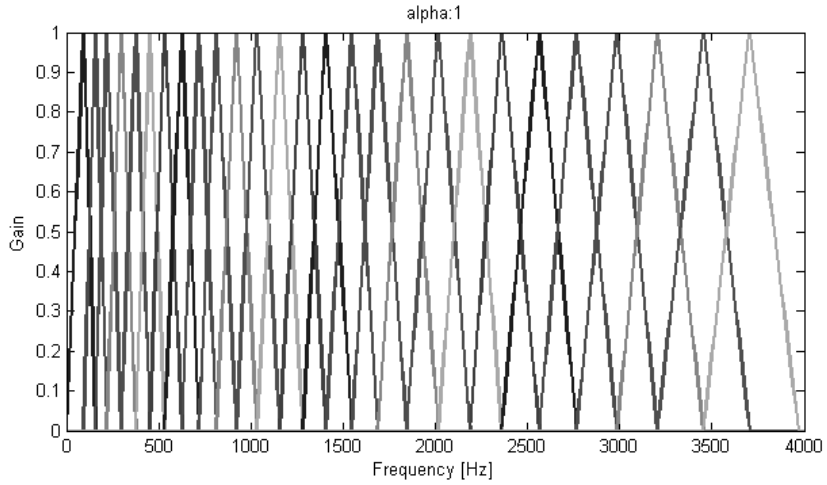

Fig. 3.5: Demonstration of filterbanks.

The next step is to apply discrete cosine transform (DCT) on the log energies E_b of the triangular bandpass filters to obtain M mel-scale cepstral coefficients.

$$C_m = \sum_{b=1}^{N} \cos\left[m(b-0.5)\pi/N\right] E_b, \quad m = 1, 2, 3, \ldots, M$$

where N is the number of triangular bandpass filters and M is the total number of mel-scale cepstral coefficients.

By taking the DCT, the higher coefficients are removed and simultaneously the spectral shape of the signal is taken. It should be noted that if Fourier transform outputs are directly used as final feature vectors, they usually will not do such a good job of returning the important information in the lower coefficients. Usually lower-order coefficients represents the spectral shape, while the higher-order coefficients are more noise-like features,

moreover in speech analysis the particular amplitude at different frequencies are less important than the general shape of the spectrum.

Finally it is often advantageous to include the time derivatives of MFCCs as new features, which shows the velocity and acceleration of each MFCC feature vectors. These extra added feature vectors are named *delta* and *double delta* features. In the simple form, the formula to compute the *delta* features is as follows:

$$d[n] = x[n+1] - x[n]$$

where $x[n]$ and $x[n+1]$ are the consecutive MFCC features of frame n, and $d[n]$ is the corresponding *delta* features for that frame. *Double delta* are obtained by replacing the MFCC features with *delta* features in the above equation.

In this study, we have used the delta features to compute a new set of feature vectors named as *shifted delta cepstral coefficients (SDC)*. SDC features have reported significant performance in language recognition systems [16]. In the following sections, the process of SDC feature extraction method will be discussed in more detail.

MFCC features are the core of many feature extraction methods in dialect and accent recognition system, including building up SDC feature vectors. But they are also used as stand alone feature vectors for dialect recognition tasks in speaker and speech recognition systems. As in [12], authors used a set of 12 mean normalized cepstral coefficients and the mean normalized log-energy to build their dialect models.

3.1.2 Feature Normalization and RASTA Filtering

A common problem with speech processing of the signals is that the characteristics of the channel or environment might vary from one session to the other. Examples are a change in frequency characteristics of a channel by changing to a new microphone or recording device. The goal in feature normalization is to reduce the effects of these irrelevant information in final extracted feature vectors from speech signals.

The frequency characteristics of communication channels are assumed to be fixed or only show slow variations in time so if the speech representation is considered invariant of these slow changes in the cepstral domain, then channel effects might not cause a serious problem.

One of the ways to deal with environmental and channel distortion effects is to consider them as a simple linear filter:

$$y[n] = x[n] * h[n]$$

where $h[n]$ is the linear filter impulse response of the channel, $*$ denotes the convolution and $x[n]$ is the representative of speech signal in discrete time domain, where, n is an integer value and denotes the sequential values of time.

Based on signal representation in frequency domain, we can write

$$Y[k] = X[k]H[k]$$

where k is integer value which denotes the sequential values of frequency. Taking the logarithm of the above equation leads to

$$log\ Y[k] = log\ X[k] + log\ H[k]$$

This equation indicates that the effect of channel distortions is added as a additive value to the signal amplitude in the log domain. By applying cepstral processing, for a given signal $x[k]$ we can write (subscript c indicates cepstral domain)

$$O_c[k] = \text{IFFT}\ \{\ln[\text{FFT}\{x[k]h[k]\}]\}$$

where FFT{.} and IFFT{.} are the fast Fourier and inverse fast Fourier transforms. So in cepstral domain we have:

$$O_c[k] = H_c[k] + x_c[k],$$

indicating that the effect of environment and channels is just adding a constant value in cepstral domain. So robustness can be achieved by estimating $h_c[k]$ and subtracting it from the observed $O_c[k]$.

But the important question rises here is that how to remove the effect of h_k from the observed cepstral features. Since speech usually varies at a faster rate than acoustic environment impulse response, the channel impulse response is reasonably considered as constant over a short amount of time. By taking the average of $O_c[k]$ over a short amount of time yields to:

$$\overline{O}_c = H_c + \overline{x}_c$$

where $\overline{h}_c$ and $\overline{x}_c$ are the short-time means of $h_c[k]$ and $x_c[k]$, respectively. By assuming that the short-time mean of speech cepstrum is zero[1], previous equation yields to $\overline{O}_c = h_c$. Therefore the clean speech cepstrum will be given as

$$\hat{O}_c[k] = O_c[k] - \overline{O}_c$$

.

Subtracting the short-time cepstral mean removes the direct-current (DC) component from the cepstral coefficients while returning the fast varying components. Conceptually, this idea indicates that the cepstral coefficients are being high-pass filtered. This technique is known as cepstral mean subtraction.

The basic idea behind the RASTA filtering comes from the fact that human hearing perception is not sensitive to slow variations of speech signals, so that the aim in RASTA filtering is to keep only the information that are due to speech sound. In RASTA processing, first the spectral coefficients of the speech signal are compressed by a non-linear compression rule. This non-linear compression may be the logarithm of the spectral coefficients. Then, the compressed values are band-passed filtered by a filter with a sharp spectral zero at the zero frequency. This filter removes the slow changes of the speech signal. The band-pass filter is usually an infinite impulse response, *IIR*, filter with the transform function [19]:

$$H(z) = 0.1z^4 \, \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.98z^{-1}}$$

The high-pass portion of the above filter removes the effect channel noise, where as the low-pass portion smooth the fast spectral changes appeared in the frame-to-frame analysis of speech signal. After band-pass filtering, the spectral coefficients are expanded by a non-linear function which is usually an exponential function. More information about RASTA processing can be found [19].

[1]In the very short time analysis, it can be assumed that the number of times the signal goes up is equal to the number of times signal goes down in frequency axis with the same amplitude in both directions

In this study, one of our experiments relates to include and exclude RASTA processing in cepstral processing. We report the results in the experimental chapter of this thesis.

3.1.3 Shifted Delta Cepstra Coefficients

Feature vector extraction method for dialect or accent identification systems is typically performed by constructing a feature vector at frame time t consisting of cepstra and delta cepstral coefficients. As previous studies have shown significant improvements can be achieved by using *shifted delta cepstra* (SDC) feature vectors created by stacking delta cepstra coefficients computed across multiple speech frames [20]. Fig. 3.6 shows how the SDCs are computed over a speech frame. Four parameters determine the SDC feature extraction, *N, D, P* and *K*. *N* is the number of cepstral coefficients computed at frame *t*, *D* represents the time advance and delay for delta computation, *P* is the time shifted between consecutive blocks and *K* represents the number of blocks whose delta coefficients are concatenated to from the final feature vector.

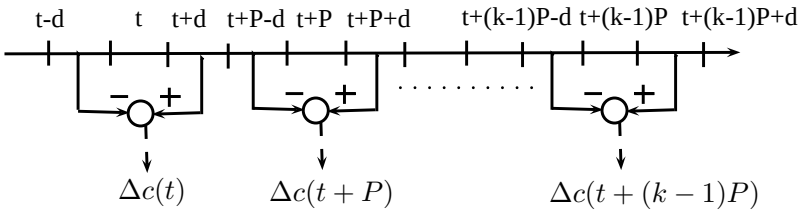

Fig. 3.6: Computation of the SDC feature vector at frame t for parameters *N, D, P* and *K*.

SDCs add more time-related information of the signal to feature vectors. Already delta cepstral coefficients includes time-relationship between consecutive frames, however SDCs can be seen as a compact representation of these time-relationships in one feature vector. Mathematically, KN parameters are used for each SDC feature vector, as compared with $2N$ for conventional cepstra and delta-cepstra feature vectors. The final vector at frame time t is given by the concatenation of all the $\Delta c\,(t + ip)$, where

$$\Delta c\,(t) = c\,(t + ip + d) - c\,(t + ip - d)$$

$$i = 0, 1, \ldots, T - 1$$

where T is the total number of frames in speech signal.

SDCs have been the main feature vectors of many works in dialect and accent recognition (in general language recognition) field. One of the recent successful studies is the NIST LRE 2011 Language Recognition System [21] of MIT Lincoln laboratory, where many state-of-the-art techniques such as

feature nuisance attribute projection (fNAP) [22] and *vocal tract length normalization* (VTLN)[1] have been applied on SDCs to create the final feature vectors for recognition process.

[1]It will be described in the following sections.

3.1.4 Vocal Tract Length Normalization

Automatic dialect and accent recognition systems should be able to deal with inter-speakers variabilities caused by physiological differences between speakers, such as gender and differences in vocal tract length. Vocal tract length normalization (VTLN) is inspired from the fact that vocal tract length varies across different speakers from 18 cm in males to approximately 13cm in females [24]. VTLN is a technique to compensate these variations among different speakers or utterances by warping the frequency axis of spectral features so that observations become more similar across all speakers in terms of VTL. This procedure reduces speaker-dependent variations in formant frequencies by a simple linear warping of the frequency axis.

Several approaches have been proposed in literature to find warping factors for vocal tract length normalization [24, 25]. The most common method is based on choosing a warp factor that gives the maximum likelihood (ML) criterion through a grid search over a range of warp factors [24]. Then warp factors will be chosen to maximize likelihoods from a reference model trained from Gaussian mixture models (GMMs) or hidden Markov models (HMMs).

However, there are some other approaches which find warp factors by considering correlation between laryngeal size and vocal tract length as stated in [43]. In this approach, a joint distribution of pitch and warp factors is estimated during training as $P(\nu|F_0)$. This distribution denotes the probability of warp factor ν given the mean of pitch over a speaker speech frames and it is used to select the most probable warp factor given a speaker's average pitch. While formant frequencies might be good indicators of vocal tract length, accurate formant extraction is difficult - especially in noisy signals. On the other hand, formant frequencies are not directly proportional to VTL as discussed in [26].

Maximum Likelihood Warp Factor Estimation

We followed a maximum likelihood (ML) approach to estimate the warp factors using mixture of multivariate Gaussians (GMM) model. This enables warp factor selection to be moved entirely into front end processing by varying the spacing and width of the filter-banks and keeping the speech spectrum unchanged. For example, in order to compress speech signal in frequency domain, we can compress the filter-bank frequencies to stretch the signal frequency scale while the frequency scale of speech signal remains the same. Regarding ML estimation, for a speaker i, let X_i^{ν} be spectral observation feature vectors with a frequency axis scaled with warp factor ν. Given observed data X_i and a reference GMM spectral model λ, the probability of

warp factor ν can be described in terms of acoustic likelihoods:

$$P\left(\nu|X_i,\lambda\right) = \frac{P\left(X_i^{\nu}|\lambda\right)}{\Sigma_{\nu'} P\left(X_i^{\nu'}|\lambda\right)}$$

Then the optimal warp factor is estimated by a grid search over a range of ν values typically between 0.88 to 1.12 with increment 0.02,

$$\hat{\nu_i} = \underset{i}{argmax}\ P\left(\nu|X_i^{\nu},\lambda\right)$$

This equation indicates that optimal warp factor for each observed utterance X_i^{ν} is selected so as to maximize its likelihood in the reference GMM. In this report ν is used to modify the mel-scaling used to compute filter-bank centers, as follows [30]

$$f_{mel} = 2595 * log_{10}\frac{(1.0+f)}{700\nu}$$

Algorithm

Below is the algorithmic procedure of computing warp factors which we use [24]:

1. Divide the training dialects into two sets, training (T) and aligning (A) subsets.
2. Compute unwarped features for the training set.
3. Train a GMM λ_T using the unwarped features of the training set.
4. For each utterance in the alignment set compute the likelihoods for each warp factor by matching it against the trained GMM from the previous step.
5. Select the optimal warping factor for each utterance i in the set alignment to maximize $P\left(X_i^{\nu}|\lambda_T\right)$
6. Swap the sets, and iterate this process of training a GMM with half of the data, and then finding the best warping factor for the second half.
7. Repeat this process until there is no significant change in $\nu's$ between two consecutive iteration.

8. Finally, for the test utterances, compute the feature vectors for different warp factors and select the one which gives the maximum likelihood once it is matched to the normalized GMM computed from training procedure.

VTLN has been a main component in most of state-of-the-art dialect and accent recognition systems. As in [36] which the focus is on Arabic dialect identification, or in [27] in which the CMLLR method is combined with VTLN to observe the effects on the performance of the recognition system.

3.1.5 Front-end Configuration

One of the important issues in dialect and accent recognition systems is the order in which the feature processing components appear in the recognition system. Fig. 3.7 demonstrates how feature processing is usually performed in dialect and accent recognition systems. We also built the front-end part of our baseline dialect recognition systems based on the configuration shown in this figure, however as we will show in experimental chapter, in one of our experiments feature normalization order is shifted in this process to see its effect on the overall recognition performance.

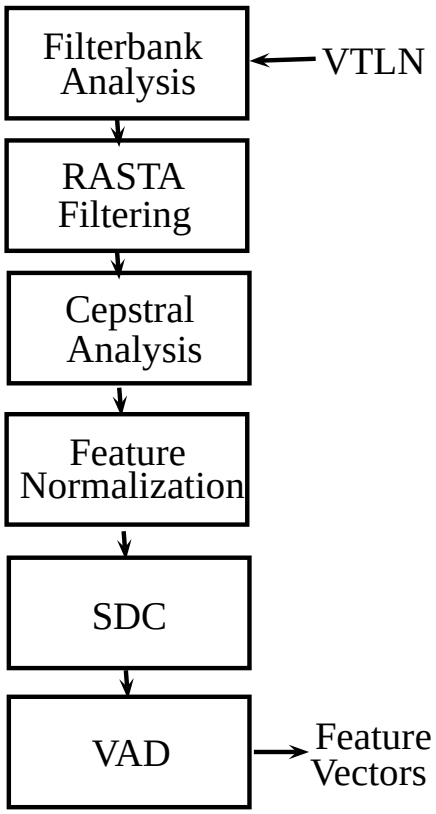

Fig. 3.7: The order in which components appear in feature processing part.

3.2 Back-end Processing

3.2.1 Multivariate Gaussian Mixture Models

Gaussian mixture model (GMM) is a statistical method used to model speaker (in our case dialect) specific features. It consists of a number of individual Gaussians to provide multi-modal density representation for each model. In pattern recognition applications, GMMs are used to generate speaker (dialect) models and also to match different patterns against the trained models.

In fact many phenomenon can be described by Gaussians *pdf*, i.e that their occurrences follow the behavior of Gaussians[1]. Multi-variative Gaussian distribution is represented as

$$P(x|\theta_m) = \frac{1}{(2\pi)^{p_m/2} det(\Sigma_m)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_m)^S \Sigma_m^{-1}(x-\mu_m)\right)$$

where S denotes the transpose operation, $x = x[0], x[1], \ldots, x[N-1]$ represents $N-1$ independent observations from a mixture model, $\theta_m = (\mu_m, \Sigma_m)$ represents the mean vector and covariance matrix of the m_{th} component, and p_m represents the m_{th} mixture weight. In creating models for dialects or accents, we have a set of observations (our feature vectors), $x[0], x[1], \ldots, x[N-1]$ and we look for a Gaussian, or better to say Gaussian mixture models, which statistically describe these observations.

But the important question is that how we can statistically describe a set of observations since there are not any prior information about the statistical distributions such as their mean or variance?

In order to answer this question, we form a *likelihood* function:

$$L(X;\theta) = \prod_{i=0}^{N} p(x_i;\theta)$$

Where θ is estimated by *Maximum Likelihood Estimation (MLE)* approach:

$$\hat{\theta}_{MLE} = arg\max_{\theta} L(X;\theta)$$

a Gaussian Mixture Model which consists of k components is formulated as

$$p(x) = j_1 N(x;\mu_1,\sigma_1) + j_2 N(x;\mu_2,\sigma_2) + \cdots + j_i N(x;\mu_i,\sigma_i)\ ; \sum_{i=1}^{k} j_i = 1,\ j_i >= 0$$

[1]Gaussians have rare probabilities on the tails, and most of the events' probabilities happen around the mean of phenomenon.

In this equation, θ is the statistical parameters of the individual component i which are components weight, j_i, components mean, μ_i and components variance, σ^2.

A two dimensional Gaussian Mixture Model is shown in Fig. 3.8. One of the properties of Gaussians is that most of the data are centered around their mean value, so that rare events are unlikely to happen.

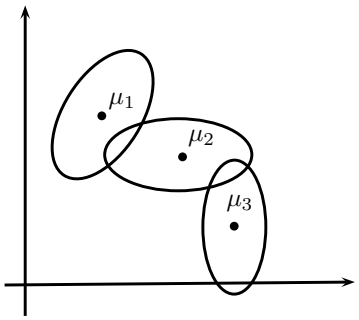

Fig. 3.8: Demonstration of a two dimensional Gaussian Mixture Model.

Maximum Likelihood Estimation of Parameters of GMM

The approach of estimating the parameter θ, which is also known as training a specified model, is based on *expectationmaximization* (EM) algorithm. This technique applies an iterative procedure to find maximum likelihood or alternatively *maximum a posteriori* (MAP) estimates of parameters in statistical models [28]. The algorithm follows the rules given below:

1. Make initial guess of parameters, c_i, μ_i, σ_i. This is usually done by use of k-means algorithm [29].

2. Knowing parameters of θ, find probability of sample x_i belonging to j^{th} component.

$$p(y[i] = j \mid x[i]; \theta) \quad \text{for } i = 1, 2, \ldots, N \Rightarrow \text{no. of observations}$$
$$\text{for } j = 1, 2, \ldots, M \Rightarrow \text{no. of components}$$

3. $$c_j^{new} = \frac{1}{N} \sum_{i=1}^{N} p(y[i] = j \mid x[i]; \theta)$$

4. $$\mu_j^{new} = \frac{\sum_{i=1}^{N} x_i . p(y[i]=j \mid x[i];\theta)}{\sum_{i=1}^{N} p(y[i]=j \mid x[i];\theta)}$$

5. $$(\sigma_j^2)^{new} = \frac{\sum_{i=1}^{N} (x_i - \hat{\mu_i})^2 . p(y[i]=j \mid x[i];\theta)}{\sum_{i=1}^{N} p(y[i]=j \mid x[i];\theta)}$$

6. Go back to (2) and repeat until convergence, usually 10 iteration is enough.

$y[i]$ is defined as $y[i] = 1$ if $x[i]$ belong to component 1, $y[i] = 2$ if $x[i]$ belong to component 2, and so on.

3.2.2 Universal Background Modeling

In dialect recognition systems, the *UBM* is a dialect-independent *Gaussian mixture model* (*GMM*) trained with speech samples from a large set of dialects to represent general dialect characteristics. By considering a test case as X, the basic hypothesis test in dialect recognition systems can be written as follows:

$P(X|M_i)$: This denotes the likelihood of X coming from hypothesized model i.

In order to decide to which model the test case X belongs (here two models are considered, M_0 denotes the correct hypothesis model and M_1 denotes the incorrect hypothesis.), the following test should be evaluated:

$$\frac{P(X|M_0)}{P(X|M_1)} = \phi$$

So that If ϕ is greater than a threshold, then we will accept this test and report that the test case is coming from the first model, M_0.

Mathematically, M_0 is characterized by a model denoted as Ω_{hyp} that is trained from the feature vectors of X. For example, we could assume that a Gaussian mixture model best represents the distribution of feature vectors for M_0 so that Ω_{hyp} will be represented as a set of mean vectors and covariance matrices and weights of the feature vectors. Alternatively, we could characterized the M_1 by a model denoted as $\Omega_{\overline{hyp}}$. Then the likelihood ratio statistic is $\frac{P\left(X|\Omega_{hyp}\right)}{P\left(X|\Omega_{\overline{hyp}}\right)}$.

Often the log-likelihood of this statistic is used for testing the hypothesizes as follows:

$$llr = \log\left(P\left(X|\Omega_{hyp}\right)\right) - \log\left(P\left(X|\Omega_{\overline{hyp}}\right)\right)$$

While the model Ω_{hyp} can be estimated by using the training samples, the model $\Omega_{\overline{hyp}}$ is less well-defined since it should represent the entire space of all hypothesizes. The concept of *UBM* is defining and modeling $\Omega_{\overline{hyp}}$. *UBM* is constructed by pooling out the data from all dialects training utterances and training one universal model [23].

The main advantage of this strategy is that we build a single dialect-independent model once and then use it for all hypothesized test cases in the experiments.

In this study, UBMs are constructed per language, i.e. for each language available in our corpus, one UBM is constructed. However, the second option

could be build a unique UBM for all three languages available by pooling out all training utterances of dialects.

3.2.3 Model Adaptation

Unlike the standard maximum likelihood approach for training both target and non-target models, construction of UBMs leads to a more efficient way in training the models which is so called adaptation. As Fig. 3.9 demonstrates, the core idea behind adaptation is to update the well-trained parameters of the target and non-target models in the UBM [23].

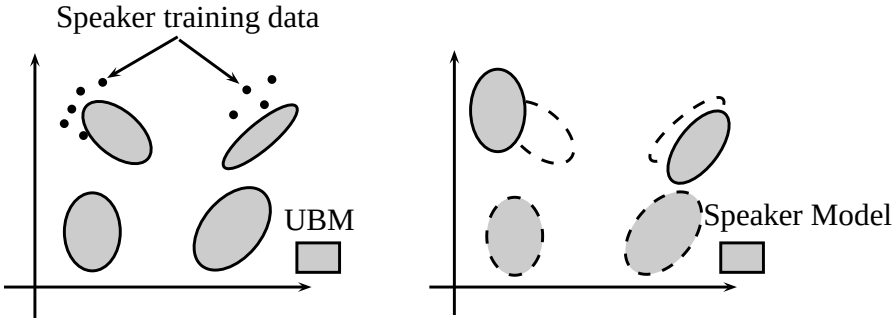

Fig. 3.9: The adapted mixture parameters are derived using the statistics of the new data and the UBM mixture parameters.

Adaptation process provides a tighter coupling between the dialects model and UBMs, which not only produces better performance than decoupled models, but also as discussed later in the next section, allows for a fast-scoring technique.

Like the EM algorithm, the adaption consists of two steps of estimation process [23]. In the first step, sufficient statistics are estimated for each mixture in the UBM and in the second step these new sufficient statistics are coupled with the old sufficient statistics from the UBM mixture parameters to create parameters of the adapted target model .

Let's have a look at the mathematics behind adaptation process. Given the UBM and the training vector from the hypothesized model, $X = [x_1, x_2, \ldots, x_T]$, for each mixture i in the UBM, we calculate

$$P\left(i|x_t\right) = \frac{j_i P_i\left(x_t\right)}{\sum_{i=1}^{M} j_i P_i\left(x_t\right)}$$

where T denotes total number of frames, $j_i and j_v$ represents the mixture weights at corresponding index, $P\left(i|x_t\right)$ represents the probability of frame x_t given the mixture probability, M denotes the number of mixtures, and finally $P\left(i|x_t\right)$ represents the probability of mixture i given the frame x_t. By

using this probability, *sufficient statistics* are computed as

$$n_i = \sum_{i=1}^{T} P\left(i|x_t\right)$$

$$E_i\left(x\right) = \frac{1}{n_i} \sum_{i=1}^{T} P\left(i|x_t\right) x_t$$

$$E_i\left(x^2\right) = \frac{1}{n_i} \sum_{i=1}^{T} P\left(i|x_t\right) x_t^2$$

As shown in above formula, sufficient statistics are computed from the models training data. n_i represents the posterior probability of the mixture i, so that it is called *count* moment. $E_i(x)$ represents the first order moment, which equivalently indicates the expectation value of the i_{th} mixture coming from speech frames. $E_i\left(x^2\right)$ is the second order moment, which represents the variance the probabilities of i_{th} mixture coming from speech frames. Applying the sufficient statistics, the adapted parameters for mixture i in the UBM are computed as

$$\hat{\omega_i} = \left[\alpha_i^{\omega} n_i/T + \left(1 - \alpha_i^{\omega}\right)\omega_i\right]\gamma$$

$$\hat{\mu_i} = \alpha_i^{m} E_i\left(x\right) + \left(1 - \alpha_i^{m}\right)\mu_i$$

$$\hat{\sigma_i^2} = \alpha_i^{v} E_i\left(x^2\right) + \left(1 - \alpha_i^{v}\right)\left(\sigma_i^2 + \mu_i^2\right) - \hat{\mu_i^2}$$

The parameters $[\alpha_i^v, \alpha_i^m, \alpha_i^\omega]$ control the balance between the old and new sufficient statistics. These are data-dependent adaptation coefficients which are defined as:

$$\alpha_i^{\rho} = \frac{n_i}{n_i + r^{\rho}}$$

Where r^ρ is a fixed *relevance factor* for parameter ρ. As an example in language recognition system r^ρ is considered as a number between 6 to 16. The use of parameter-dependent relevance factors allows tuning of different adaptation rates for the weights, means, and variances so that one of the experiments of this research is to analyze the impact of different r^ρ on the overall performance of the system. The scale factor, γ, is computed over all adapted mixture weights to ensure they sum to unity.

Universal background modeling together with adaptation process have been used the baseline system of many studies in language, dialect, and

speaker recognition systems, so that researches usually compare their designed system performance with this baseline system. We also built our baseline recognition system based on the idea of adaptation and universal background modeling presented in [30].

3.2.4 Fast Scoring Method

As discussed earlier, adaptation of models parameters based on UBMs allows for a faster method to evaluate the scores of the models. The fast scoring method is based on two facts, first, when dealing with large GMMs, only a few of the components in the mixtures have significant impact on the log-likelihood values and secondly, during the adaptation process, feature vectors which are close to particular components in the UBM will also be close to the corresponding component in the target model. Using these two observations, for each feature vector, first we determine the top H scoring components in the UBM and compute UBM likelihood using only those top H components. Next, we score the test vector against only the corresponding H components in the adapted dialect or accent model to get the utterance's likelihood [23]. The pseudocode of this method is as follows:

For each frame $t = 1, 2, \ldots, T$

For each components $k = 1, 2, \ldots, M$ compute

$P_{ubm}(k, t) = j_k \times N(x_t|\mu_k, \Sigma_k)$

End

sort $P_{ubm}(k)$ across t and select the top H scores

Where $N(x_t|\mu_k, \Sigma_k)$ is calculated as follows:

$$P(x_t|M_i) = \frac{1}{T}\sum_{k=1}^{M} j_k \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_k|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x_t - \mu_k)^S \Sigma_k^{-1}(x_t - \mu_k)\}$$

Where T is the total number of frames, S denotes the transpose operation, M the is number of Gaussian components, D is the dimension of feature vectors and j_k is the components weights.

3.2.5 Identity Vectors

Identity vector (i-vector) systems have been the current state-of-the-art language recognition systems [31]. The idea behind the i-vector systems is to consider between-class variability in the space of model parameters (we have different model parameters for each language or dialect) and also the within-class variability (parameters of a specific language or dialect can change from utterance to utterance because of differences in channels, speakers, reverberation, etc) in one global variable namely as *total variability space*. But how this global variability, between-class variability and within-class variability variability, can be modeled? Regarding the GMM-UBM adaptation process, we can create *supervectors* from GMM models for each utterance. Supervectors are high and fixed dimensional data built from concatenation of all GMM components means (Fig. 3.10). Conceptually the GMM supervector can be consider as a mapping between an utterance and a high-dimensional vector [32].

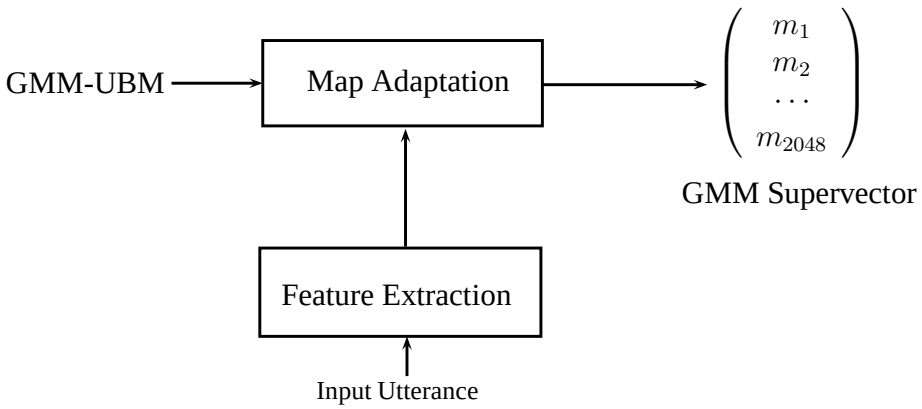

Fig. 3.10: GMM supervector systems.

Given a dialect utterance supervector, and based on the *joint factor analysis* (JFA) approach [33], a statistical method is used to demonstrate observed variability in form of lower number of unobserved variables named *factors*. In this model, the dialect-dependent supervector, μ_i is defined as

$$\mu_i = m + Vy_i + Ux_i + Dz_i,$$

where

- Vector m is a dialect independent supervector (from the pre-trained UBM).

- Matrix V is the *eigen-voice matrix*.
- Vector y contains the *dialect factors*. Assumed to have $N(0,1)$ prior distribution.
- Matrix U is the *eigen-channel matrix*.
- Vector x contains the *channel factors*. Assumed to have $N(0,1)$ prior distribution.
- Matrix D is the *residual diagonal matrix*.
- Vector z contains the speaker (dialect) specific *residual factors*, assumed to have $N(0,1)$ prior distribution.

The above equation shows that each dialect model can be ideally split into a set of independent objects which accounts for the global variability matrix which we already discussed. [35] found that the subspaces U and V are not completely independent, therefore a combined *total variability* space was introduced in [34]. In this approach, there is no distinction between the between-class variability and within-class variability in the dialect dependent GMM supervector μ_i. Therefore, the new dialect dependent GMM supervector is rewritten as follows:

$$M = m + \mathrm{T}w$$

, where m is the dialect-dependent supervector, T is called the total variability matrix and w corresponds to the i-vectors which controls an eigendimension of the T. For a given utterance, w is defined by its posterior distribution conditioned to the *Baum-Welch* statistics. The posterior distribution is a Gaussian distribution and the mean of this distribution corresponds to the i-vectors.

Given the UBM and the training vector from the hypothesized model, $X = [x_1, x_2, \ldots, x_T]$, for each mixture $i = 1, 2, ..., M$ in the UBM, we calculate

$$P\left(i|x_t\right) = \frac{w_i P_i\left(x_t\right)}{\sum_{j=1}^{M} w_j P_j\left(x_t\right)}$$

By using this probability, *sufficient statistics* are computed as

$$n_i = \sum_{i=1}^{T} P\left(i|x_t\right)$$

$$E_i(x) = \frac{1}{n_i} \sum_{i=1}^{T} P(i|x_t)\, x_t$$

,

$$E_i\left(x^2\right) = \frac{1}{n_i} \sum_{i=1}^{T} P(i|x_t)\, x_t^2$$

First order Baum-Welch statistics are also needed in order to estimate w.

$$\tilde{E}_i = \sum_{t=1}^{T} P(i|x_t)(x_t - m_i)$$

Where m_i is the mean of UBM mixture component i.

Finally, the i-vector for a given utterance is computed as

$$w = (I + \mathrm{T}^t \Sigma^{-1} n(u) \mathrm{T})^{-1} \mathrm{T}^t \Sigma^{-1} \tilde{E}(u)$$

Where $n(u)$ is a diagonal matrix of dimension $MT \times MT$ with diagonal elements $n_i I$ $(i = 1, \ldots, M)$. $\tilde{E}(u)$ is a supervector of dimension $MT \times 1$ obtained by stacking all first-order Baum-Welch statistics $\tilde{E}_i$ for a given utterance u. Σ is a diagonal covariance matrix of dimension $MT \times MT$ computed during factor analysis training and it models the residual variability not estimated by the total variability matrix T.

There are two popular methods for scoring the estimated i-vectors. The first method uses *linear discriminant analysis* (LDA) and *cosine scoring* approach as described in [35]. In the second approach, the distributions of i-vectors for individual dialect is modeled by Gaussian distributions with a full covariance matrix shared across all dialects. For a given i-vector w corresponding to a test utterance, the log-likelihood for each dialect is computed as [42]:

$$\ln p(w|d) = -\frac{1}{2} w^S \Sigma^{-1} w + w^S \Sigma^{-1} \mu_d - \frac{1}{2} \mu_d^S \Sigma^{-1} \mu_d$$

where μ_d is the mean of dialect d and Σ is common covariance matrix and S denotes the transpose operation.

3.2.6 Evaluation Metrics

In this study, the performance results are reported based on three evaluation metrics, identification error rate (IDerror), equal error rate (EER) and minimum detection cost function (minDCF). Identification error rate is a binary classification between the correct and incorrect dialects hypothesises and is computed by dividing number of incorrect hypotheses over all test utterances. Equal error rate is the point on a DET curve where the false acceptance rate and false rejection rate are equal. Lower equal error rate indicates the better performance in the recognition systems. DET curve or *detection error tradeoff* curve is the plots of the error rates for binary classification systems which plots false rejection rates vs. false acceptance rates. Finally *minimum detection cost function* is defined as a weighted sum of the miss and false alarm error probabilities or equivalently minDCF represents the expected cost of making a detection decision [38]. Mathematically, minDCF is defined as the minimum value of C_{Det} from the equation below

$$C_{Det} = C_{Miss}(P_{Miss}|Target)P_{Target}+$$

$$...C_{FalseAlarm}(P_{FalseAlarm}|NonTarget)P_{NonTarget}$$

where C_{Miss} and $C_{FalseAlarm}$ are relative costs of detection errors, P_{Target} is the prior probability of the target hypothesis, and $P_{NonTarget}$ is $1 - P_{Target}$. The value of C_{Miss} and $C_{FalseAlarm}$ are considered 1 in this study.

Chapter 4

Experimental Results

In this chapter, the results of different experiments conducted in this study are presented. The experiments are divided into two parts, experiments focus on tuning the parameters of the front-end processing, and the experiments focus tuning the parameters of back-end processing. All the experiments reports are drawn in three tables corresponding to three languages used in this study. Analysis of experiments are discussed in the last section of this chapter.

The evaluation metrics considered for reporting the results, are based on identification error rate (IDerror), equal error rate (EER) and minimum value of the detection cost function (minDCF).

4.1 Data Preparation

We used the *CallFriend* corpus [2] to test the performance of our designed system. This database is collection of unscripted conversations for 12 languages. It includes two dialects for each language available. The audio files have been recorded over telephone lines so that two channels (stereo audio files) have been used for recordings, one for the interviewer and the other for participants.

The corpus has three different partitions, each organized for specific tasks, *training* folder is for training the dialect models, *development* folder is used for testing the dialect models and parameter tunings and, finally, *evaluation* folder is for reporting the final test accuracies of the system.

We selected three languages from the corpus. These languages are English, Mandarin and Spanish. Each of these languages has two dialects: *North* and *South* for English, *Mandarin* and *Taiwanese* for Chinese, and *Caribbean* and *Non-Caribbean* for Spanish. Each audio file samples in the corpus are

about 30 minutes chunks. For purpose of this work, each of the speech utterances available has been partitioned into 30 seconds length. Table 1 shows some statistics regarding the corpus. In this work, we used the train subset utterances for training the dialect models, and evltest subset utterances for reporting the system performance.

Table 4.1: Number of data available in CallFriend Corpus dialects after splitting into 30s length.

Dialects	EN/EN_SOUTH	MA/MA_T	SP/SP_CAR
train	4425/3975	3416/4151	4110/4145
devtest	4406/4427	4037/4658	4445/4685
evltest	4082/4146	4375/4256	4172/4122

4.2 Results

In this section, experimental results are given. Each experiment has its own run-specific configuration. But a couple of common specifications among all experiments are as follows, the N-d-P-K parameters of SDC method are 7-1-3-7, respectively, creating feature vectors of dimension 49, C0 included in cepstral coefficients, by feature normalization we mean cepstral mean variance normalization (CMV) method [39], number of iteration in adaptation process is 1 and dialects log-likelihood scores are calibrated with multi-class logistic regression method from FoCal Multi-class Toolkit [6].

Voice activity detection experiment

In voice activity detection (VAD) experiment, the aim is to observe the effect of VAD threshold[1] on the dialect recognition performance. In this Experiment, non-speech frames are removed from SDC features with 20%, 50%, 70% and 100% thresholds. Specifically, 100% VAD threshold means that all non-speech frames are removed from the SDC feature vector.

The run-specific configurations of this experiment are as follow: feature normalization method is applied on MFCC features, RASTA filter is on,

[1]The output of the voice activity detection is a set of 0's and 1's for each the SDC frame. 0's indicate the non-speech parts and 1's indicate the speech parts for each frame. In order to decide whether a whole SDC frame is a speech or non-speech frame, we define a threshold based on the number of 1's (speech parts) in the frame. If it is larger than a defined threshold, we consider the frame as a speech frame. For example, for a defined 20% threshold, if the number of speech parts (1's) divided by total number of speech and non-speech parts is greater than 20%, the frame is considered as speech frame and its cepstral features are kept in the final feature vector.

VTLN is off, the relevance factor in adaptation process is 6, and number of GMM components is 2048.

Tables 2, 3 and 4 demonstrate the results of this experiment for English, Mandarin and Spanish languages, respectively. The first row of each table corresponds to the experiment in which, non-speech frames are first removed prior to cepstral analysis, in contrast to remove non-speech frames from the final SDC feature vectors in VAD experiment. This experiment is referred as *No-VAD* experiment in future usages.

Table 4.2: Experimental results for English language in *VAD* experiment.

Experiments	EER	IDerror	minDCF
No-VAD	17.45%	35.63%	0.1194
20% VAD threshold	17.75%	37.67%	0.1223
50% VAD threshold	17.16%	36.15%	0.1144
70% VAD threshold	15.51%	35.32%	0.1089
100% VAD threshold	**14.48%**	**34.62%**	**0.0913**

Table 4.3: Experimental results for Mandarin language in *VAD* experiment.

Experiments	EER	IDerror	minDCF
No-VAD	17.03%	31.72%	0.1114
20% VAD threshold	14.66%	28.98%	0.0972
50% VAD threshold	13.98%	28.41%	0.0899
70% VAD threshold	13.11%	27.57%	0.0826
100% VAD threshold	**11.39%**	**27.44%**	**0.0716**

Table 4.4: Experimental results for Spanish language in *VAD* experiment.

Experiments	EER	IDerror	minDCF
No-VAD	17.69%	33.95%	0.1246
20% VAD threshold	15.87%	34.74%	0.1094
50% VAD threshold	15.84%	33.01%	0.1075
70% VAD threshold	15.26%	32.28%	0.1063
100% VAD threshold	**12.00%**	**32.25%**	**0.0742**

Mel-frequency cepstral coefficients concatenated to shifted delta cepstral coefficients

The aim of this experiment is to concatenate mel-frequency cepstral coefficients (MFCCs) to shifted delta cepstral coefficients. We refer to the experiment as *MFCCs-added-to-SDCs* in future usages.

The run-specific configurations of this experiment are as follow: 7 first MFCCs are concatenated to SDCs to from the final feature vectors creating

feature vectors of dimension 56, feature normalization is applied on MFCC features, RASTA filter is on, VTLN is off, the relevance factor in adaptation process is 6, and number of GMM components is 2048.

It should be noted that, in order to have a baseline comparison, a second experiment conducted in which SDC feature vectors are used as stand-alone features for building the dialect models. We refer to this experiment as *SDC* in future usages. It should be noted that in both of these two experiments, non-speech frames are removed from MFCC feature vectors.

Tables 5, 6 and 7 demonstrate the results of these experiments for English, Mandarin and Spanish languages, respectively.

Table 4.5: Experimental results for English language in *MFCCs-added-to-SDCs* experiments.

Experiments	EER	IDerror	minDCF
MFCCs-added-to-SDCs	**15.67%**	**33.82%**	**0.1044**
SDC	17.45%	35.63%	0.1194

Table 4.6: Experimental results for Mandarin language in *MFCCs-added-to-SDCs* experiments.

Experiments	EER	IDerror	minDCF
MFCCs-added-to-SDCs	**14.43%**	**30.09%**	**0.0941**
SDC	17.03%	31.72%	0.1114

Table 4.7: Experimental results for Spanish language in *MFCCs-added-to-SDCs* experiments.

Experiments	EER	IDerror	minDCF
MFCCs-added-to-SDCs	**16.82%**	**33.42%**	**0.1181**
SDC	17.69%	33.95%	0.1246

Feature Normalization applied on shifted delta cepstral coefficients

The aim of this experiment is to apply CMV feature normalization method on final SDC feature vectors. We refer to the experiment as *SDC-FN* in future usages.

The run-specific configurations of this experiment are as follow: feature normalization is applied on the final shifted delta cepstral coefficients, RASTA filter is on, VTLN is off, the relevance factor in adaptation process is 6, and number of GMM components is 2048.

It should be noted that, in order to have a baseline comparison, a second experiment conducted in which CMV feature normalization is applied on

MFCC feature vectors. We refer to this experiment as *MFCC-FN* in future usages. It should be noted that in both of these two experiments, non-speech frames are removed from SDC features with 100% VAD threshold.

Tables 8, 9 and 10 demonstrate the results of these experiments for English, Mandarin and Spanish languages, respectively.

Table 4.8: Experimental results for English language in *SDC-FN* experiment.

Experiments	EER	IDerror	minDCF
SDC-FN	14.97%	35.01%	0.0982
MFCC-FN	**14.48%**	**34.62%**	**0.0913**

Table 4.9: Experimental results for Mandarin language in *SDC-FN* experiment.

Experiments	EER	IDerror	minDCF
SDC-FN	12.49%	29.00%	0.0776
MFCC-FN	**11.39%**	**27.44%**	**0.0716**

Table 4.10: Experimental results for Spanish language in *SDC-FN* experiment.

Experiments	EER	IDerror	minDCF
SDC-FN	14.96%	34.76%	0.0954
MFCC-FN	**12.00%**	**32.25%**	**0.0742**

RASTA experiment

The aim of this experiment is to turn RASTA filter off during front-end processing. We refer to the experiment as *RASTA-off* in future usages.

The run-specific configurations of this experiment are as follow: RASTA filter is off, feature normalization is applied on MFCC features, VTLN is off, the relevance factor in adaptation process is 6, and number of GMM components is 2048.

It should be noted that, in order to have a baseline comparison, a second experiment conducted in which RASTA filter is kept on during front-on processing. We refer to this experiment as *RASTA-on* in future usages. It should be noted that in both of these two experiments, non-speech frames are removed from SDC features with 70% VAD threshold.

Tables 11, 12 and 13 demonstrate the results of these experiments for English, Mandarin and Spanish languages, respectively.

Table 4.11: Experimental results for English language in *RASTA* experiment.

Experiments	EER	IDerror	minDCF
RASTA-Off	18.31%	36.60%	**0.0982**
RASTA-On	**15.51%**	**35.32%**	0.1089

Table 4.12: Experimental results for Mandarin language in *RASTA* experiment.

Experiments	EER	IDerror	minDCF
RASTA-Off	14.99%	27.84%	0.1016
RASTA-On	**13.11%**	**27.57%**	**0.0826**

Table 4.13: Experimental results for Spanish language in *RASTA* experiment.

Experiments	EER	IDerror	minDCF
RASTA-Off	16.68%	**31.02%**	0.1161
RASTA-On	**15.26%**	32.28%	**0.1063**

Vocal tract length normalization experiment

The aim of this experiment is to apply vocal tract length normalization (VTLN) on front-end processing. We refer to the experiment as *VTLN-on* in future usages. The run-specific configurations of this experiment are as follow: VTLN is on, feature normalization is applied on MFCC features, the relevance factor in adaptation process is 6, and number of GMM components is 2048.

It should be noted that, in order to have a baseline comparison, a second experiment conducted in which VTLN is not included during front-on processing. We refer to this experiment as *VTLN-off* in future usages. It should be noted that in both of these two experiments, non-speech frames are removed from SDC features with 70% VAD threshold.

Tables 14, 15 and 16 demonstrate the results of these experiments for English, Mandarin and Spanish languages, respectively.

Table 4.14: Experimental results for English language in *VTLN* experiment.

Experiments	EER	IDerror	minDCF
VTLN-Off	15.51%	35.32%	0.1089
VTLN-On	**14.91%**	**32.93%**	**0.1007**

Table 4.15: Experimental results for Mandarin language in *VTLN* experiment.

Experiments	EER	IDerror	minDCF
VTLN-Off	13.11%	27.57%	0.0826
VTLN-On	**12.20%**	**25.82%**	**0.0770**

Table 4.16: Experimental results for Spanish language in *VTLN* experiment.

Experiments	EER	IDerror	minDCF
VTLN Off	15.26%	32.28%	**0.1063**
VTLN On	**15.16%**	**31.55%**	0.1082

Number of GMM components experiment

The aim of this experiment is to vary the number of GMM components in training the UBMs. We refer to the experiment as *GMM Component* in future usages.

The run-specific configurations of this experiment are as follow: number of GMM components vary by 256, 512, 1024 and 2048, feature normalization is applied on MFCC features, non-speech frames are removed from SDC features with 70% VAD threshold, VTLN is off and the relevance factor in adaptation process is 6.

Tables 17, 18 and 19 demonstrate the results of these experiments for English, Mandarin and Spanish languages, respectively.

Table 4.17: Experimental results for English language in *GMM Component* experiment.

Experiments	EER	IDerror	minDCF
256 GMM Components	20.68%	38.04%	0.1474
512 GMM Components	19.20%	37.85%	0.1404
1024 GMM Components	17.70%	36.74%	0.1256
2048 GMM Components	**15.51%**	**35.32%**	**0.1089**

Table 4.18: Experimental results for Mandarin language in *GMM Component* experiment.

Experiments	EER	IDerror	minDCF
256 GMM Components	16.77%	29.21%	0.1198
512 GMM Components	15.43%	28.82%	0.1060
1024 GMM Components	14.27%	27.66%	0.0941
2048 GMM Components	**13.11%**	**27.57%**	**0.0826**

Table 4.19: Experimental results for Spanish language in *GMM Component* experiment.

Experiments	EER	IDerror	minDCF
256 GMM Components	18.74%	33.56%	0.1421
512 GMM Components	17.97%	33.17%	0.1331
1024 GMM Components	16.93%	32.78%	0.1214
GMM Components-2048	**15.26%**	**32.28%**	**0.1063**

Relevance factor experiment

The aim of this experiment is to vary the value of relevance factor, r in models adaptation process. We refer to the experiment as *r-value* in future usages.

The run-specific configurations of this experiment are as follow: the value of relevance factor in adaptation process is varied by 6, 11 and 16, feature normalization is applied on MFCC features, non-speech frames are removed from SDC features with 70% VAD threshold, RASTA filter is on, and VTLN is off.

Tables 20, 21 and 22 demonstrate the results of these experiments for English, Mandarin and Spanish languages, respectively.

Table 4.20: Experimental results for English language in *relevance factor* experiment.

Experiments	EER	IDerror	minDCF
r-Value-6	15.51%	35.32%	0.1089
r-Value-11	15.61%	35.30%	0.1093
r-Value-16	15.62%	35.30%	0.1096

Table 4.21: Experimental results for Mandarin language in *relevance factor* experiment.

Experiments	EER	IDerror	minDCF
r-Value-6	13.11%	27.57%	0.0826
r-Value-11	13.02%	27.08%	0.0820
r-Value-16	13.13%	27.08%	0.0820

Table 4.22: Experimental results for Spanish language in *relevance factor* experiment.

Experiments	EER	IDerror	minDCF
r-Value-6	15.26%	32.28%	0.1063
r-Value-11	15.91%	32.40%	0.1083
r-Value-16	15.97%	32.37%	0.1087

Identity vector experiment

Our last experiment is aimed at applying identity vector (i-vector) system on the back-end processing of our dialect and accent recognition system.

The run-specific configurations of this experiment are as follow: In the front-end processing SDC feature vectors of dimension 49 corresponding to 7-1-3-7 SDC parameters are used. Then the features are normalized by VTLN and CMV normalization methods. The threshold considered for VAD is 70%.

On the other hand, in i-vector system configurations we considered 5 iterations in T-matrix training, i-vectors are of dimension 600 and 2048 GMM components are used in UBM training. We used the *train* files folder in UBM training. Moreover, in estimating the T-matrix, we used the sufficient statistics of CallFriend corpus for capturing the total variabilities. As an

example, in case of English language, the T-matrix is built up from 1000 sufficient statistics of Mandarin and Spanish chunk utterances.

Table 23 demonstrates the result of this experiments for each of the three languages available.

Table 4.23: Experimental results of *identity vector* experiments for each of the three languages.

Languages	EER	IDerror	minDCF
English	11.31%	28.43%	0.0807
Mandarin	8.41%	23.56%	0.0543
Spanish	9.12%	28.24%	0.0569

4.3 Analysis of Experiments

In this section, we have a closer look at the experimental results summarized above. Then, we continue by analyzing the outputs of the experiments. The analysis consists of mentioning the contribution(s) of each each experiment, as well as comparing the the outputs of related experiments, together with reviewing literature which is related to the each experiment. In experiments analysis, we selected the identification error rate and minimum detection cost function in order to compare the relevant experiments. In case no logical conclusion is not drawn from these too metrics, we use the equal error rate to compare the relevant experiments. However, we could use other evaluation metrics pairs for this purpose, but it would not pose any difference in experiments analysis.

Voice activity detection experiment

Fig. 4.1 and Fig. 4.2 demonstrate the plot of minimum detection cost function and identification error rate at different VAD thresholds, respectively. As these plots indicate, performance results are improved as VAD threshold increases, so that at threshold 100% best performance is resulted for each of three languages available. As already explained, thresholds mean that how much non-speech frames are removed from the feature frames. So that 100% VAD threshold means that all non-speech frames are removed and the recognition process is based on only speech frames.

Moreover, it can be observed from these figures that, Mandarin language shows lower identification error rate and minimum detection cost function compared to the other two languages. This indicates that there are more distinctive characteristics in Mandarin dialects than the other languages.

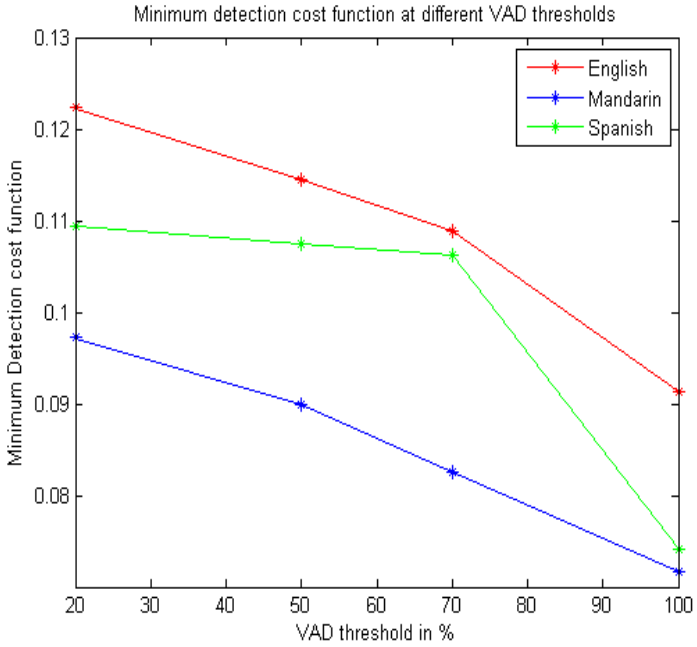

Fig. 4.1: Minimum detection cost function at different VAD thresholds.

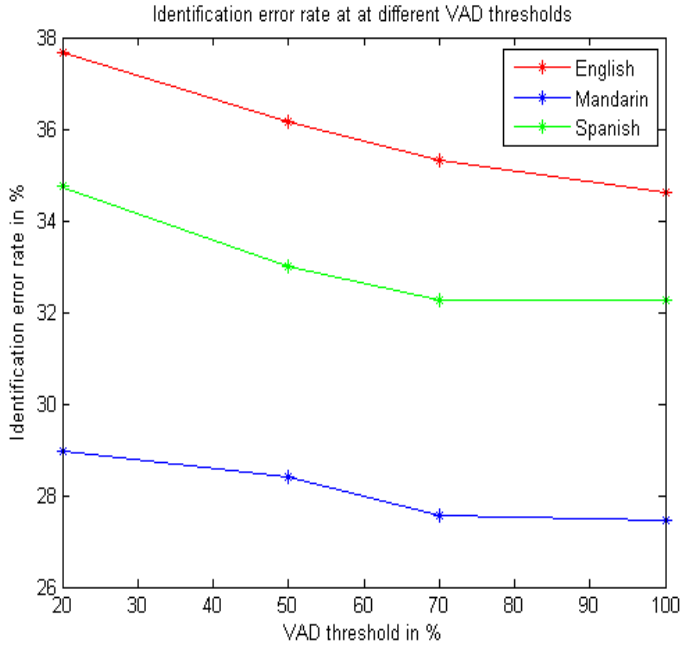

Fig. 4.2: Identification error rate at different VAD thresholds.

VAD experiments vs. No-VAD experiment

In Tables 4.24 and 4.25, we compared the minimum detection cost function and identification error rates of *VAD* experiment and *No-VAD* experiment, respectively. In the case of VAD experiment, we selected the 100% threshold corresponding to the best system performance in VAD experiment. As these tables demonstrate, *VAD* experiment win over *No-VAD* experiment both in minDCFs and identification error rate. In case the of minDCFs, the minimum and maximum relative improvements are approximately 0.02, 0.04, respectively, and in the case of identification error, the minimum and maximum relative improvements rate are 2% and 4%, respectively.

VAD experiments conclude that as more non-speech frames are removed from the frames, the lower error rates are achieved. As discussed in the theory chapter of this thesis, dialects bear very little similarity in their phonemes, so that it makes it difficult even to truly discriminate between dialects of the same language. It can be inferred that as more non-speech frames are removed from the signal, more important and discriminative features remain for dialects modelling.

Table 4.24: Minimum detection cost function comparison between 100% threshold *VAD* experiment with *No-VAD* experiment.

Languages	VAD experiment	No-VAD experiment
English	0.0913	0.1194
Mandarin	0.0716	0.1114
Spanish	0.0742	0.1246

Table 4.25: Identification error rates comparison between 100% threshold *VAD* experiment with *No-VAD* experiment.

Languages	VAD experiment	No-VAD experiment
English	34.62%	35.63%
Mandarin	27.44%	31.72%
Spanish	32.25%	33.95%

MFCCs-added-to-SDCs experiment Vs. SDC experiment

In Tables 4.26 and 4.27, we compared the minimum detection cost function and identification error of *SDC* experiment vs *MFCCs-added-to-SDCs* experiment, respectively. As these tables demonstrate, *MFCCs-added-to-SDCs* experiment win over *SDC* experiment both in minDCFs and identification error rate. In case of minDCFs, the minimum and maximum improvements are approximately 0.01 and 0.02, respectively and in case of

identification error rate they are 1% and 2 respectively.

Table 4.26: Minimum detection cost function comparison between *MFCCs-added-to-SDCs* experiment with *SDC* experiment.

Languages	MFCCs-added-to-SDCs experiment	SDC experiment
English	0.1044	0.1194
Mandarin	0.0941	0.1114
Spanish	0.1181	0.1246

Table 4.27: Identification error rate comparison between *MFCCs-added-to-SDCs* experiment with *SDC* experiment.

Languages	MFCCs-added-to-SDCs experiment	SDC experiment
English	33.82%	35.63%
Mandarin	30.09%	31.72%
Spanish	33.42%	33.95%

SDC-FN experiment vs. MFCC-FN experiment

In Tables 4.28 and 4.29, we compared the minimum detection cost function and identification error of these two experiments, respectively. As these tables demonstrate, feature normalization applied on MFCC feature vectors before forming the SDC features outperforms slightly feature normalization applied on final SDC feature vectors. Specially, in the case of Spanish dialects, this improvement is clearer than the other two languages.

As discussed in [39], most of the normalization methods are applied on the Mel-frequency cepstral coefficient (MFCC) speech features. However, to best knowledge of author, no attempts have been made to see the effect of feature normalization on SDCs or other features formed from mel-cepstral coefficients. The reason behind why feature normalization applied on MFCC feature vectors before forming the SDC features outperforms slightly feature normalization applied on final SDC feature vectors, might originate from the fact that feature normalization methods are used in speaker and dialect recognition systems to compensate the effect environmental distortions. So as these negative effects are compensated earlier during front-end processing, other components of the system will be less effected by the distortions.

Table 4.28: Minimum detection cost function comparison between feature normalization on MFCCs with feature normalization on SDCs.

Languages	MFCC-FN experiment	SDC-FN experiment
English	0.0913	0.0982
Mandarin	0.0716	0.0776
Spanish	0.0742	0.0954

Table 4.29: Identification error rate comparison between feature normalization on MFCCs with feature normalization on SDCs.

Languages	MFCC-FN experiment	SDC-FN experiment
English	34.62%	35.01%
Mandarin	27.44%	29.00%
Spanish	32.25%	34.76%

RASTA-off experiment vs. RASTA-on experiment

In Tables 4.30 and 4.31, we compare the minimum detection cost function and identification error rate of these two experiments. Concerning minDCFs and identification error rate, no clear conclusion can be made, because in case of Mandarin and Spanish languages, once the RASTA filter is on, the system performance increases, while English does not benefit from RASTA. The same condition happens in case of identification error rate. So we shift our comparison metric to equal error rate. As Tables 4.32 represents, once the RASTA filter is on, there is a considerable improvement in EER values, so that the at least 1% absolute improvement is achieved for all the three languages.

Although RASTA filtering technique has reported significant improvement in automatic speech recognition (ASR) systems [5], but no previous works independently concentrated on RASTA filtering of dialect utterances in dialect and accent recognition task. To sum up, RASTA filtering improves the recognition performance of the dialect and accent recognition system.

Table 4.30: Minimum detection cost function comparison between *RASTA-off* experiment with *RASTA-on* experiment.

Languages	RASTA-off experiment	RASTA-on experiment
English	0.0982	0.1089
Mandarin	0.1016	0.0826
Spanish	0.1161	0.1063

Table 4.31: Identification error rate comparison between *RASTA-off* experiment with *RASTA-on* experiment.

Languages	RASTA-off experiment	RASTA-on experiment
English	36.60%	35.32%
Mandarin	27.84%	27.57%
Spanish	31.02%	32.28%

Table 4.32: Equal error rate comparison between *RASTA-off* experiment with *RASTA-on* experiment.

Languages	RASTA Filter off	RASTA Filter on
English	18.31%	15.51%
Mandarin	14.99%	13.11%
Spanish	16.68%	15.26%

VTLN-on experiment vs. VTLN-off experiment

In Tables 4.33 and 4.34, we compared the minimum detection cost function and identification error rate of these two experiments, respectively. In general, VTLN has improved both minDCFs and identification error rate in all the languages available. But considerable improvements are observed in EER values, so that the minimum and maximum absolute improvements are 2% and 3%, respectively. Although VTLN considerably increases the process time of the recognition process, but it has been a main component in state-of-the art dialect and accent recognition systems [36, 27]. Furthermore, our result show that how VTLN could benefit dialects which have less discriminative characteristics, such as English and Spanish dialects in our experiments.

Table 4.33: Minimum detection cost function comparison between *VTLN-off* experiment and *VTLN-on* experiment.

Languages	VTLN-off experiment	VTLN-on experiment
English	0.1089	0.1007
Mandarin	0.0826	0.0770
Spanish	0.1063	0.1082

Table 4.34: Identification error rate comparison between *VTLN-off* experiment and *VTLN-on* experiment.

Languages	VTLN-off experiment	VTLN-on experiment
English	35.32%	32.93%
Mandarin	27.57%	25.82%
Spanish	32.28%	31.55%

Table 4.35: Equal error rate comparison between *VTLN-off* experiment and *VTLN-on* experiment.

Languages	VTLN-off experiment	VTLN-on experiment
English	18.31%	15.51%
Mandarin	14.99%	13.11%
Spanish	16.68%	15.26%

Number of GMM components experiment

Fig. 4.3 and Fig. 4.4 demonstrate the plot of minimum detection cost function and identification error rate for four different number of GMM components, respectively. As these plots indicate, performance results are constantly improved as number of GMM components increases, so that at 2048 value the best performance is resulted for each of the languages available.

As the number of GMM components increases, it takes a longer time for the systems to report the performance. So that some researchers as in [27] preferred to use 512 GMM components for models training, in some other works as in [30], 2048 GMM components have been used. Our research contributes that as the number of GMM components increases, significant improvements can be achieved with the cost of increased time complexity.

Moreover, it can be observed that Mandarin language shows better results compared to the other two languages. This indicates that there are more distinctive characteristics in Mandarin dialects than in the dialects of the other two languages.

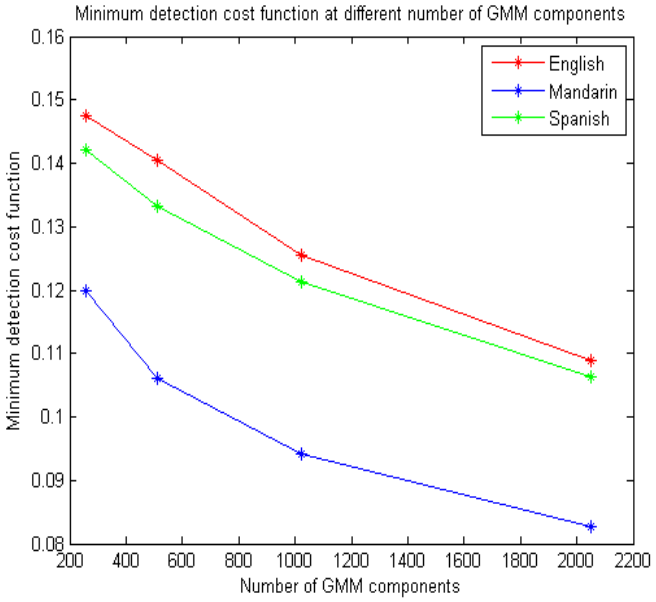

Fig. 4.3: Minimum detection cost function at four different number of GMM components.

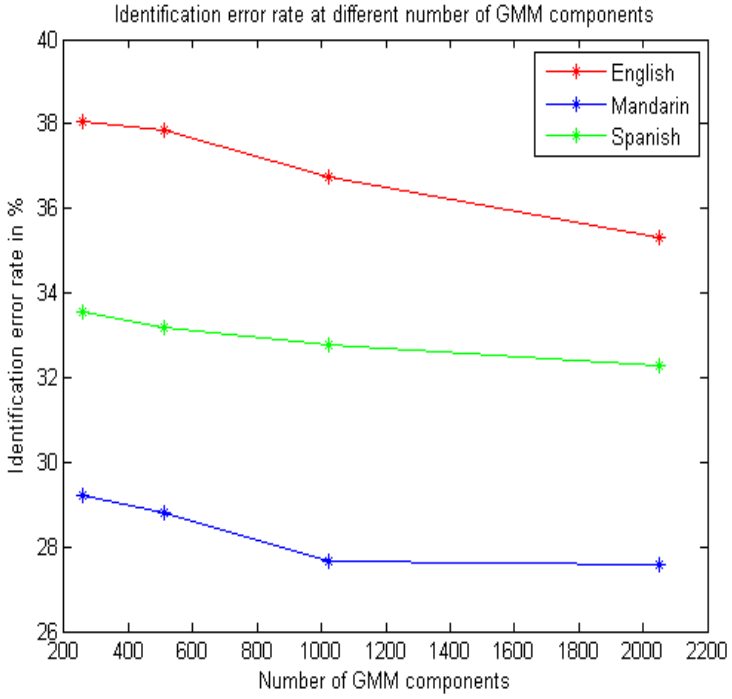

Fig. 4.4: Identification error rate at four different number of GMM components.

Relevance factor experiments

Fig. 4.5 and Fig. 4.6 demonstrate the plot of minimum detection cost function and identification error rate for three different relevance factors, respectively. As these plots indicate, no considerable changes are observed in neither minimum detection cost function values nor identification error rates.

One of the interesting property observed is that Mandarin language is still showing better performance over other two languages. To the best knowledge of the author, no specific experiments were conducted in the literatures to mention the optimized value of relevance factor in adaptation process. So one of the contributions of this study is that in the tasks of dialect and accent recognition, the system performance is not sensitive to the value of r during the adaptation process.

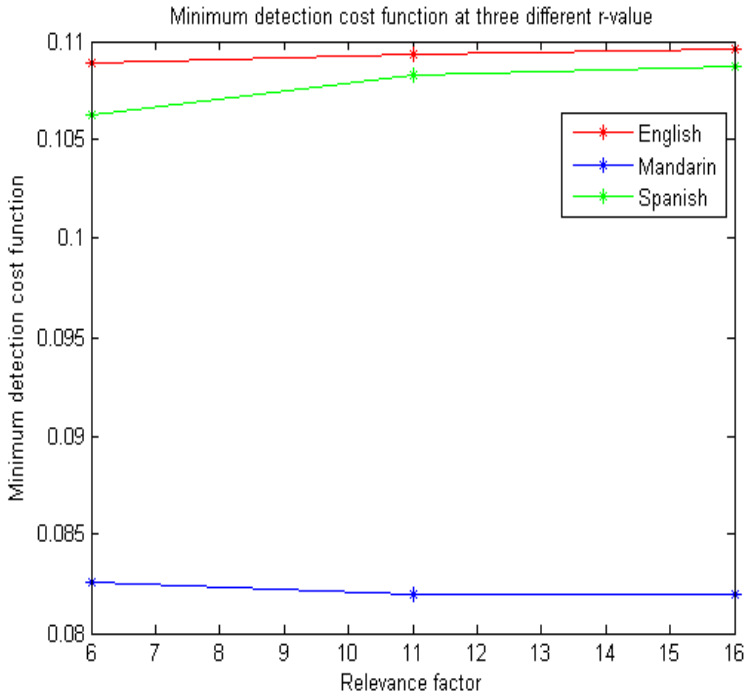

Fig. 4.5: Minimum detection cost function at three different relevance factors.

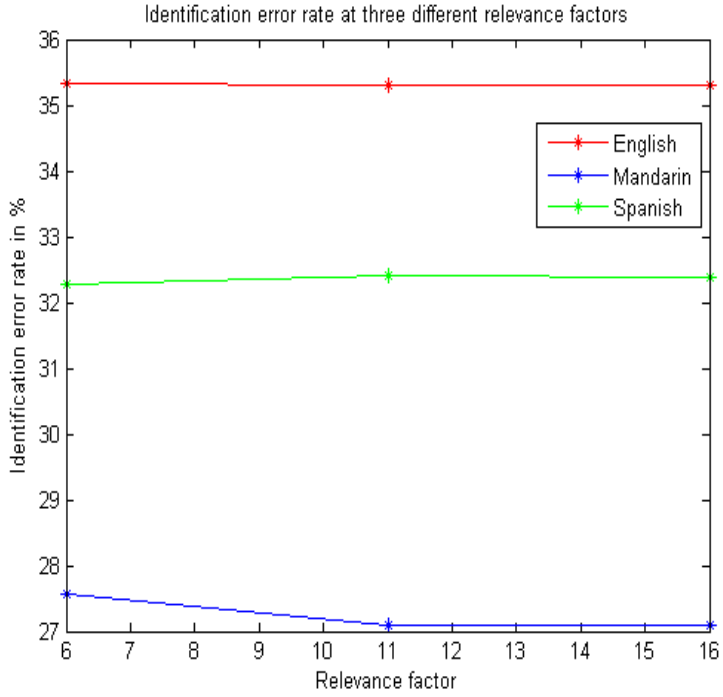

Fig. 4.6: Identification error rate at three different relevance factors.

Chapter 5

Summary and Conclusion

As discussed in the introduction, the goal we considered in this research was to tune different parameters in front-end and back-end processing to see their effects in the performance of the system. Another goal that we were looking for in this research was to make our system to get closer to the state-of-the-art dialect and accent recognition systems by adding VTLN and i-vector systems to the front-end and back-end processing part of our designed system.

In the previous chapter, we conducted a number of comparative experiments. A summary of our research achievements are listed below

1. The best performance achieved refers to i-vector system with at least 3% absolute improvements both in equal error rate and identification error rate.

2. Mandarin dialects show the best distinctiveness characteristics comparing to the other two language dialects. Spanish dialects and English dialects are placed in the next positions, respectively.

3. As the number of GMM components increases, the performance of the system is drastically improved.

4. Performance of the system was less sensitive to relevance factor variations in the adaptation process so that no significant changes are observed in the evaluation metrics.

5. Voice activity detection applied on final SDC feature vectors help in improving the results, so that at 100% VAD threshold best performance is achieved regarding VAD experiment.

6. Concatenating mel-frequency cepstral coefficients to shifted delta cepstral coefficients to from the feature vectors, improves the performance of the system.

7. Vocal tract length normalization improves the overall performance of the system.

8. Applying cepstral mean variance feature normalization on mel-cepstral coefficients outperforms applying this normalization on final SDC feature vectors.

9. RASTA filtering of the features helps improving the performance results.

10. Regarding the sensitivity of the results to tuning parameters, the system is more sensitive to changes in number of GMM components and VAD thresholds than the other parameters. VTLN, RASTA filtering, concatenation of mel-cepstral coefficients to SDC feature vectors, and feature normalization appear in next orders, respectively. On the other hand, the least sensitivity belongs to relevance factor variations.

However, we did not evaluate the performance of our dialect and accent recognition system on other available dialect speech corpuses, and from this viewpoint, it was one of the limitations we faced during this study. Regarding future work, we are going to expand the analysis of the system on other available corpuses such as available dialects in the Miami corpus or Latin American Spanish accent speech database. Furthermore, we want to focus on the control parameters of the i-vector system, and how to improve the performance of i-vector system. It would also be interesting to couple our dialect and accent recognition system with ASR systems.

References

[1] Spanish in spain. http://www.enforex.com/language/spanish-spain.html. Accessed: 13/12/2012.

[2] Callfriend corpus. In *Linguistic Data Consortium*, 1996.

[3] Adrian Akmajian, Richard A. Demers, and Robert M. Harnish. *Linguistics: an introduction to language and communication"*. MIT Press, 2 edition, 1984.

[4] Fadi Biadsy, Hagen Soltau, Lidia Mangu, and Jiri Navratil Julia. Discriminative phonotactics for dialect recognition using context-dependent phone classifiers, 2010.

[5] Louis Boves and Johan de Vethd. Comparison of channel normalisation techniques for automatic speech recognition over the phone. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 4, pages 2332 –2335 vol.4, oct 1996.

[6] Niko Brummer. Focal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition scores-tutorial and user manual. 2007.

[7] Gang Liu and John L. Hansen. A systematic strategy for robust automatic dialect identification. In *EUSIPCO2011*, pages 2138–2141, 2011.

[8] Gang Liu, Yun Lei, and John H.L. Hansen. Dialect identification: Impact of differences between read versus spontenous speech. In *EUSIPCO-2010*, pages 49–53, 2010.

[9] John Nerbonne. Linguistic variation and computation. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 3–10, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[10] Jurg Strassler. *Introduction to Phonetics and Phonology*. Springer, 2010.

[11] Pedro A. Torres-Carrasquillo, Douglas E. Sturim, Douglas A. Reynolds, and Alan McCree. Eigen-channel compensation and discriminatively trained gaussian mixture models for dialect and accent recognition. In *INTERSPEECH*, pages 723–726. ISCA, 2008.

[12] Tingyao Wu, Jacques Duchateau, Jean-Pierre Martens, and Dirk Van Compernolle. Feature subset selection for improved native accent identification. *Speech Commun.*, 52(2):83–98, February 2010.

[13] Yonghua Xu, Jian Yang, and Jiang Chen. Methods to improve gaussian mixture model for language identification. In *Proceedings of the 2010 International Conference on Measuring Technology and Mechatronics Automation - Volume 02*, ICMTMA '10, pages 656–659, Washington, DC, USA, 2010. IEEE Computer Society.

[14] Marc A. Zissman. Comparison of four approaches to automatic language identification of telephone speech. *Speech and Audio Processing, IEEE Transactions on*, 4(1):31, jan 1996.

[15] W. Zue, Timothy J. Hazen, and Timothy J. Hazen. Automatic language identification using a segment-based approach. In *Proc. Eurospeech*, pages 1303–1306, 1993.

[16] Bocchieri Bielefeld, Language identification using shifted delta cepstrum. In *Fourteenth An- nual Speech Research Symposium*, 1994.

[17] Jacob Benesty, M. Mohan Sondhi, and Yiteng Arden Huang. *Springer Handbook of Speech Processing*. Springer, 2007.

[18] Elizabeth K. Hanson, David R. Beukelman, Jana Kahl Heidemann, and Erin Shutts-Johnson. The impact of alphabet supplementation and word prediction on sentence intelligiblity of electronically distorted speech. *Speech Communication*, 52(2):99–105, 2010.

[19] Hynek Hermansky and Nelson Morgan. RASTA processing of speech. In *IEEE Transactions on Speech and Acoustics*, volume 2, pages 587–589, October 1994.

[20] Mary A. Kohler and Michael P. Kennedy. Language identification using shifted delta cepstra. In *Circuits and Systems*, 2002. *MWSCAS-2002*. The 2002 45th Midwest Symposium on, volume 3, pages III 69–72 vol.3, aug. 2002.

[21] Elliot Singer, Pedro Torres-Carrasquillo, Douglas Reynolds, Alan McCree, Fred Richardson, Najim Dehak, and Doug Sturim. The mitll nist lre 2011 language recognition system. In *Odyssey 2012 The Speaker and Language Recognition Workshop*, pages 4994–4997, march 2011.

[22] William M. Campbell, Douglas E. Sturim, Douglas Reynolds, and Alex Solomonoff. SVM based speaker verification using a gmm supervector kernel and nap variability compensation. In *in Proceedings of ICASSP*, 2006, pages 97–100.

[23] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted gaussian mixture models. In *Digital Signal Processing*, page 2000, 2000.

[24] Li Lee and R.C. Rose. Speaker normalization using effcient frequency warping procedures. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings.*, 1996 IEEE International Conference on, volume 1, pages 353–356 vol. 1, may 1996.

[25] Sankaran Panchapagesan and Abeer Alwan. Frequency warping for vtln and speaker adaptation by linear transformation of standard mfcc. *Com- put. Speech Lang.*, 23(1):42–64, January 2009.

[26] Ellen Eide and Herbert Gish. A parametric approach to vocal tract length normalization. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings.*, 1996 IEEE International Conference on, volume 1, pages 346–348 vol. 1, may 1996.

[27] Wade Shen and D. Reynolds. Improved gmm-based language recognition using constrained mllr transforms. In Acoustics, Speech and Signal Pro- cessing, 2008. *ICASSP 2008. IEEE International Conference on*, pages 4149–4152, April 2008.

[28] Todd K. Moon. The expectation-maximization algorithm. *Signal Processing Magazine, IEEE*, 13(6):47–60, nov 1996.

[29] Xin guang Li, Min feng Yao, and Wen tao Huang. Speech recognition based on k-means clustering and neural network ensembles. In Natural Computation (ICNC), *2011 Seventh International Conference on*, volume 2, pages 614–617, july 2011.

[30] Pedro A. Torres-Carrasquillo, Douglas A. Reynolds, and P. Gleason. Dialect identification using gaussian mixture models. In *ISCA*, pages 757–760, 2004.

[31] David Martnez Gonzlez, Luks Burget, Luciana Ferrer, and Nicolas Schef- fer. ivector-based prosodic system for language identification. In *ICASSP*, pages 4861–4864. IEEE, 2012.

[32] William M. Campbell, Douglas E. Sturim, and Douglas Reynolds. Support vector machines using gmm supervectors for speaker verification. *Signal Processing Letters*, IEEE, 13(5):308–311, may 2006.

[33] Patrick Kenny, Joint factor analysis versus eigenchannels in speaker recognition. Audio, *Speech, and Language Processing, IEEE Transactions on*, 15(4):1435–1447, may 2007.

[34] Najim Dehak, Reda Dehak, Patrick Kenny, Niko Brummer, Pierre Ouellet, and Pierre Dumouchel. Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification. In *INTERSPEECH'09*, pages 1559–1562, 2009.

[35] Najim Dehak, Patrick J. Kenny, Rda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 2010.

[36] Murat Akbacak, Dimitra Vergyri, Andreas Stolcke, Nicolas Scheffer, and Arindam Mandal. Effective arabic dialect classifcation using diverse phonotactic models. In *INTERSPEECH'11*, pages 737–740, 2011.

[37] Murat Akbacak, Dimitra Vergyri, Andreas Stolcke, Nicolas Scheffer, and Arindam Mandal. Effective arabic dialect classification using diverse phonotactic models. In *INTERSPEECH11*, pages 737-740, 2011.

[38] The 2005 NIST language recognition evaluation plan, pp. 1–6, NIST, 2005.

[39] Md Jahangir Alam, Pierre Ouellet, Patrick Kenny, and Douglas O'Shaughnessy. Comparative evaluation of feature normalization techniques for speaker verification. In *Proceedings of the 5th international conference on Advances in nonlinear speech processing, NOLISP'11*, pages 246–253, Berlin, Heidelberg, 2011. Springer-Verlag.

[40] Andrea DeMarco, Stephen J. Cox, Iterative classification of regional British Acccents in i-vector space, In *Proceedings of the Symposium on Machine Learning in Speech and Language Processing (SIGML 2012)*. 2012.

[41] Chadawan Ittichaichareon, Siwat Suksri and Thaweesak Yingthawornsuk, Speech Recognition using MFCC. *International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012)*, July 28-29, 2012 Pattaya (Thailand).

[42] David Martnez Gonzlez, Oldrich Plchot, Luks Burget, Ondrej Glembek, and Pavel Matejka. Language recognition in ivectors space. In *INTER- SPEECH*, pages 861–864. ISCA, 2011.

[43] Arlo Faria and David Gelbart. Efficient pitch-based estimation of VLTN warp factors. In Proc. *Eurospeech*, 2005.

[44] Mingkuan Liu, Bo Xu, Taiyi Hunng, Yonggang Deng, and Chengrong Li. Mandarin accent adaptation based on context-independent/context-dependent pronunciation modeling. In *Proceedings of the Acoustics, Speech, and Signal Processing*, *ICASSP '00*, pages II1025–II1028, Washington, DC, USA, 2000. IEEE Computer Society.

[45] InfoTalk a multi-lingual conversational speech understanding technology. http://www.infotalkcorp.com/english/apps/index.html. Accessed : 03/01/2013.

[46] Fadi Biadsy, Julia Hirschberg, and Nizar Habash, Spoken Arabic Dialect Identification Using Phonotactic Modeling, *in Proceedings of EACL 2009 Workshop on Computational Approaches to Semitic Languages*, Athens, Greece, 2009.

[47] Umit H. Yapanel and John H. L. Hansen. A new perspective on feature extraction for robust in-vehicle speech recognition. In ISCA Proc.: Eurospeech, 2003, pages 1281–1284, 2003.

[48] Manish P. Kesarkar. Feature extraction for speech reconition. In ISCA Proc.: Eurospeech, 2003, pages 1281–1284, 2003.

[49] Douglas Shaughnessy, Speech Communication: Human and Machine. India:University Press ,2001.

[50] Lawrence R. Rabiner, Ronald W. Schafer, Digital Processing of Speech Signals. Englewood Cliffs, New Jersey: Prentice-Hall, 1978.