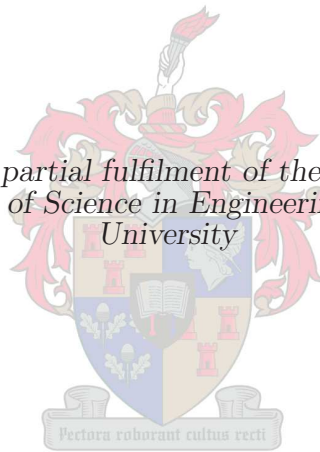


Language Identification Using Gaussian Mixture Models

by
CALVIN NKADIMENG

*Thesis presented in partial fulfilment of the requirements for the
degree of Master of Science in Engineering at Stellenbosch
University*



SUPERVISOR: Prof. T.R. Niesler
Department of Electrical & Electronic Engineering

March 2010

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March 2010

COPYRIGHT © 2010 STELLENBOSCH UNIVERSITY

ALL RIGHTS RESERVED

Abstract

The importance of Language Identification for African languages is seeing a dramatic increase due to the development of telecommunication infrastructure and, as a result, an increase in volumes of data and speech traffic in public networks. By automatically processing the raw speech data the vital assistance given to people in distress can be speeded up, by referring their calls to a person knowledgeable in that language.

To this effect a speech corpus was developed and various algorithms were implemented and tested on raw telephone speech data. These algorithms entailed data preparation, signal processing, and statistical analysis aimed at discriminating between languages. The statistical model of Gaussian Mixture Models (GMMs) were chosen for this research due to their ability to represent an entire language with a single stochastic model that does not require phonetic transcription.

Language Identification for African languages using GMMs is feasible, although there are some few challenges like proper classification and accurate study into the relationship of languages that need to be overcome. Other methods that make use of phonetically transcribed data need to be explored and tested with the new corpus for the research to be more rigorous.

Opsomming

Die belang van die Taal identifiseer vir Afrika-tale is sien 'n dramatiese toename te danke aan die ontwikkeling van telekommunikasie-infrastruktuur en as gevolg 'n toename in volumes van data en spraak verkeer in die openbaar netwerke. Deur outomaties verwerking van die ruwe toespraak gegee die noodsaaklike hulp verleen aan mense in nood kan word vinniger-up ”, deur te verwys hul oproepe na 'n persoon ingelichte in daardie taal.

Tot hierdie effek van 'n toespraak corpus het ontwikkel en die verskillende algoritmes is gementeer en getoets op die ruwe telefoon toespraak gegee. Hierdie algoritmes behels die data voorbereiding, seinverwerking, en statistiese analise wat gerig is op onderskei tussen tale. Die statistiese model van Gauss Mengsel Modelle (GMM) was gekies is vir hierdie navorsing as gevolg van hul vermo te verteenwoordig 'n hele taal met 'n enkele stogastiese model wat nodig nie fonetiese tanscription nie.

Taal identifiseer vir die Afrikatale gebruik GMM haalbaar is, alhoewel daar enkele paar uitdagings soos behoorlike klassifikasie en akkurate ondersoek na die verhouding van TALE wat moet oorkom moet word. Ander metodes wat gebruik maak van foneties getranskribeerde data nodig om ondersoek te word en getoets word met die nuwe corpus vir die ondersoek te word strenger.

Acknowledgements

I would like to thank God for giving us faith to persevere. I would especially like to thank my supervisors Dr. T.R. Niesler for his guidance and support while working with me on this thesis, as well as his patience. A special thanks to my family for encouraging me to carry-on in difficult times. Furthermore, I would like to thank my colleagues in the DSP lab for their motivation.

Contents

List of Figures	vii
List of Tables	x
1 Introduction	1
1.1 Project Motivation	2
1.2 System description	2
1.3 Outline	2
2 Mathematical Fundamentals	3
2.1 Front-end Processing (Feature Extraction)	3
2.1.1 Cepstral Analysis	3
2.1.2 FilterBank Analysis	4
2.1.3 Mel-Frequency Cepstral Coefficients	5
2.1.4 The Discrete Cosine Transform (DCT)	6
2.1.5 MFCC_E, MFCC_E_D and MFCC_E_D_A	7
2.1.6 Shifted Delta Cepstra (SDC)	8
2.2 Statistical Models	8
2.2.1 One-dimensional Gaussian Distribution	9
2.2.2 Two-Dimensional Gaussian Distribution	9
2.2.3 N-Dimensional Gaussian Distribution	10
2.2.4 Diagonal Covariance Approximation	11
2.2.5 Maximum-likelihood parameter estimates for a Gaussian distribution	12
2.2.6 Gaussian Mixture Models	13
2.2.7 EM Algorithm	13
2.2.8 Hidden Markov Models	15
2.2.9 Training HMMs	16
2.3 Summary	20
3 A Survey Of Multi-Lingual Speech Corpora	21
3.1 OGI_TS	21
3.1.1 Selection Of Languages	22
3.1.2 Data Collection Process	22
3.1.3 Corpus Validation And Annotation	23
3.2 GLOBALPHONE	23

3.2.1	Selection Of Languages	24
3.2.2	Data Collection Process	24
3.2.3	Corpus Validation and Annotation	25
3.3	CALLFRIEND	25
3.3.1	Selection Of Languages	25
3.3.2	Data Collection Process	25
3.3.3	Validation and Annotation	25
3.4	The Sub-Saharan Language Corpus	26
3.4.1	Selection Of Languages	26
3.4.2	Data Collection Process	29
3.4.3	Data Evaluation	30
3.5	Data Preparation for the Sub-Saharan Language Corpus	30
3.5.1	Naming convention	30
3.5.2	File format conversion	30
3.5.3	Silence pruning and speech file segmentation	31
3.5.4	Set division	31
3.6	Summary	33
4	GMM LID Systems	35
4.1	Maximum Likelihood Classification Approach	35
4.1.1	Parameterization	36
4.1.2	Training	36
4.1.3	Experimental Results	37
4.2	GMM Tokenization Approach	38
4.2.1	Parameterization	39
4.2.2	Training	40
4.2.3	Experimental Results	40
4.3	UBM Approach	42
4.3.1	Parameterization	43
4.3.2	Training	43
4.3.3	Experimental Results	43
4.4	SDC Approach	45
4.4.1	Parameterization	45
4.4.2	Training	45
4.4.3	Experimental Results.	45
4.5	Conclusion	46
5	System Development and Evaluation	47
5.1	The generic system development and evaluation process	47
5.2	Diagonal Covariance GMMs	49
5.2.1	MFCC_E	50
5.2.2	MFCC_E_D	50
5.2.3	MFCC_E_D_A	50
5.2.4	Experimental results	50
5.3	Full Covariance GMMs	52
5.4	Shifted Delta Cepstra	53

CONTENTS

vii

5.5	Universal Background Model	54
5.6	GMM to HMM conversion	56
5.7	Error analysis	59
5.8	Summary	64
6	Summary and conclusions	65
7	Recommendations and future work	67
	bibliography	69

List of Figures

2.1	Homomorphic filtering of a speech signal.	4
2.2	Triangular filter spread over a frequency spectrum according to Mel scale.	6
2.3	The MFCC feature vector extraction process.	7
2.4	The different MFCC vector structures.	8
2.5	The SDC feature construction process for $k=3$ and $P=3$	9
2.6	Unimodal Gaussian of a single random variable with $\sigma=1$ and $\mu=5$	10
2.7	Scattergram of two random variables with $\mu_x = 10, \mu_y = 10, \sigma_x = 0.1, \sigma_y = 10$ and $\rho = 0$	11
2.8	Bimodal Gaussian Histogram.	13
2.9	Illustration of a single state HMM.	16
4.1	LID system based on maximum likelihood classification using GMMs.	36
4.2	Feature vector processing in a GMM tokenizer.	39
4.3	A GMM Tokenization system followed by language dependent models.	39
4.4	Single GMM tokenizer configuration used in [13].	40
4.5	Average error rate obtained in [13] when using a single tokenizer for 12-language identification.	41
4.6	Average error rate as a function of multiple tokenizers.	42
4.7	LID system based on UBM Adaptation.	43
4.8	Variation of the UBM LID performance with respect to the number of mixtures selected during likelihood computation.	44
4.9	Comparison of the performance of a GMM LID system using conventional cepstral features and another using SDC features with respect to number of mixtures.	46
5.1	Block diagram of the system development process.	48
5.2	Block diagram of the testing process.	49
5.3	Block diagram of the grammer used in the recognition process.	49
5.4	Performance of diagonal covariance LID system using MFCC_E, MFCC_E_D and MFCC_E_D_A parameterisation.	52

5.5	Performance of full covariance LID system using MFCC_E and MFCC_E_D parameterisation.	53
5.6	Performance of SDC systems based on 10 MFCCs and 13 MFCCs.	54
5.7	Block diagram of the UBM system development and evaluation process.	58
5.8	The accuracy in percentage of identifying the language correctly in Universal Background Model system.	59
5.9	Diagrammatic representation of the GMM to HMM model conversion process. The possible paths that the transition from one state to the next are shown with the interconnecting lines labeled a_{ij} indicating the state it is coming from i and the state it is going to j . Only the transitions of state two and three are shown to avoid clutter, however the paths for states four and five adhere to the same principles.	62

List of Tables

3.1	Composition of the OGLTS corpus.	21
3.2	Composition of the GlobalPhone corpus.	23
3.3	Composition of CALLFRIEND corpus.	26
3.4	Frequency of occurrence of various language families in the SSLC corpus.	29
3.5	Composition of the SSLC corpus before data preparation.	29
3.6	File distribution in the SSLC corpus after data preparation.	32
4.1	Division of the OGLTS Corpus into Training and Test sets.	37
4.2	Experimental results for various language pairs (% Error).	38
4.3	Experimental results when using 10 languages (% Error).	38
4.4	Comparison of the performance of Standard GMM (Zissman) versus UBM system.	44
5.1	The accuracy in percentage of identifying the language correctly for a diagonal covariance system using the MFCC_E, MFCC_E_D and MFCC_E_D_A parameterisation.	51
5.2	The accuracy in percentage of identifying the language correctly for a full covariance system using the MFCC_E and MFCC_E_D parameterisation.	53
5.3	The accuracy in percentage of identifying the language correctly in Shifted Delta Cepstra systems.	55
5.4	The accuracy in percentage of identifying the language correctly in Universal Background Model systems.	57
5.5	Comparison of the accuracy of identifying the language correctly between GMM and GMM to HMM systems.	58
5.6	Confusion matrix for best performing LID system. Columns indicate correct language, while rows indicate the classification made by the LID system.	60
5.7	Identification accuracy within language families.	63

Chapter 1

Introduction

The importance of having an efficient automatic language identification (LID) system dealing with large databases of languages is to allow for further processing to be carried out on the hypothesised languages. To date, a lot of research has been carried out on LID systems that concentrate mostly on European and a few Asian languages and have to a large extent ignored African languages. It is thus desirable to develop an LID system for the sub-Saharan region of Africa that will add to the minimal research that has been conducted on this subject in the region.

In order to achieve this, feasibility issues such as the availability of resources had to be taken into consideration. The resources that were considered are a speech corpus from the region and the processing power required to automate this task. For the purposes of this research a language corpus was compiled and an Intel dual-core 1.800 GHz desktop computer was used to develop various LID systems using the HTK tools [17].

The performance of the developed system was determined for the compiled corpus, and conclusions were drawn.

Languages generally differ from one another with respect to their short term acoustics. These differences are not only caused by different phonemes employed in the languages, but also by the different manner in which these phonemes are realised in those languages [6].

Progress has been made in speech recognition by using methods such as Hidden Markov Models (HMMs) and artificial Neural Network (NNs) to model short-term acoustics. These models have proven to be robust with respect to factors like speaker differences for successful speech recognition.

The same approaches have been applied to language ID in various forms, however now with the aim of differentiating between entire languages and not the sounds making up a particular language. One approach is to model an entire language using a single stochastic model. In order to identify the language of an unknown utterance, it is decoded with each of these models in turn. The language of the model with the highest likelihood is taken to be the language of utterance.

Experience has shown that representative phoneme models perform better than those relying on a single stochastic model per language. The main disadvantage of the phonemic approach is that it requires phonemically labeled data in each of the target languages.

1.1 Project Motivation

The aim of this work is to develop and test language identification systems for the specific case of the languages found in southern Africa. To do so established algorithms in the field will be surveyed, and selected candidates implemented.

1.2 System description

The system development consists of three important steps: Data preparation, system training and system evaluation. Data preparation includes all the pre-processing, such as preparing the raw speech data to be in a format that is compatible and appropriate for the tools that will be employed in the system. The training stage includes the creation of the acoustic models, and the evaluation stage applies these models to determine their effectiveness.

All systems are trained and tested using the HTK tools.

1.3 Outline

Chapter 2 seeks to explain the fundamental signal processing and statistical principles of the algorithms that are used. Chapter 3 looks at some characteristics of language corpora that are used in LID. Chapter 4 looks at 4 approaches of LID systems that use GMMs the basic modelling method. Chapter 5 explains how the experiments were conducted and what results were obtained from these experiments.

Chapter 2

Mathematical Fundamentals

This chapter will review some signal processing principles that are important for converting the speech waveform into some form of parametric representation. Thereafter attention is given to statistical modelling by means of Gaussian Mixture Models (GMMs).

2.1 Front-end Processing (Feature Extraction)

Before statistical models can be obtained for languages, the raw speech signals must be pre-processed so as to extract features that can be used by a classification system. The use of cepstra have been particularly successful in this regard.

2.1.1 Cepstral Analysis

The speech production process can be viewed as an excitation signal $e(t)$ which is passed through a filter representing the effect of a vocal tract. For voiced sounds, the excitation is periodic and produced by the vibration of the vocal chords. For unvoiced sounds, the excitation is stochastic and due to a constriction somewhere in the vocal tract.

Assume that the vocal tract filter has an impulse response $v(t)$. Then the speech $s(t)$ can be modelled as the convolution of the excitation with the vocal tract filter impulse response:

$$s(t) = e(t) * v(t). \quad (2.1)$$

The objective of cepstral analysis is to separate the two terms on the right hand side of this equation, and hence to allow us to obtain $v(t)$ from the speech signal $s(t)$.

In the frequency domain,

$$S(f) = E(f) \cdot V(f), \quad (2.2)$$

where $V(f)$ is the frequency response of the vocal tract filter and $S(f)$ is the spectrum of the speech signal. Since $e(t)$ is periodic for voiced sounds, $E(f)$ exhibits a quickly varying ripple, which is superimposed on the more slowly varying frequency response $V(f)$.

By taking the logarithm we obtain the following relation.

$$\log S(f) = \log E(f) + \log V(f) \quad (2.3)$$

Hence the quickly and slowly varying components become additive in $\log S(f)$. In speech analysis $E(f)$ is normally separated from $V(f)$ by obtaining the Fourier transform of $\log S(f)$ and then discarding high-frequency components. Figure 2.1 illustrates this process graphically. Furthermore, $\log S(f)$ is normally approximated by means of filter-bank analysis in order to mimic the frequency sensitivity of the human ear. These steps will be described next.

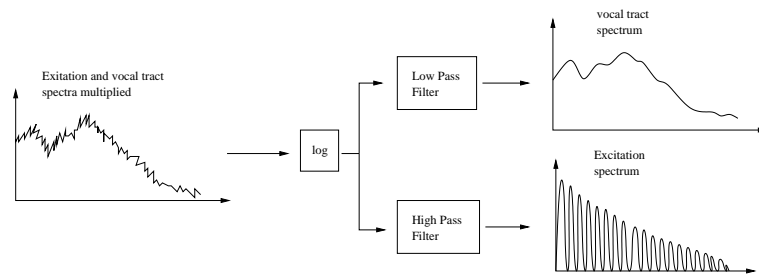


Figure 2.1: Homomorphic filtering of a speech signal.

2.1.2 FilterBank Analysis

The human auditory system is complex, and the hearing process is not fully understood, especially the brain's interpretation of the nerve signals coming from the ear. Thus a better understanding of this system could help us design better speech processing systems.

For this purpose we consider the inner part of the ear, in particular the cochlea, which is a spiral chamber filled with fluid. The spiral walls of the cochlea are made of a membrane known as the basilar membrane. The basilar membrane is stiffest near the oval window and least stiff towards the end, giving it a characteristic frequency response along its walls.

A sound enters the ear through the external canal as longitudinal air pressure waves resonating on the ear drum. This resonance causes mechanical vibrations that are transmitted to the oval window at the entrance of the cochlea, by 3 sets of bones known as the Hammer, Anvil and Stirrup. The mechanical vibrations create ripples of the fluid in the cochlea that cause the basilar membrane to

vibrate at frequencies commensurate with the input acoustic wave frequencies and at places along the basilar membrane that are associated with these frequencies. Hence, the cochlea can be modelled as a mechanical realisation of a bank of filters [11].

A filterbank is an array of bandpass filters that cover a desired portion of the frequency spectrum. It strives to isolate different frequencies within a signal; this is useful as some frequencies are deemed more important than others. Instead of arranging the band pass filters evenly over a linear frequency scale, a nonlinear frequency scale, the Mel scale, is used by speech processing algorithms to mimic the frequency sensitivity of the human ear [17]. The Mel frequency for a frequency f is given by:

$$Mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \quad (2.4)$$

Filterbanks using the Mel scale are used to compute a particular parameterisation of the cepstrum, known as Mel-Frequency Cepstral Coefficients (MFCCs).

2.1.3 Mel-Frequency Cepstral Coefficients

In order to compute Mel-frequency Cepstral Coefficients (MFCCs), the filterbank is chosen to consist of filters that are triangular in shape, and hence defined by three parameters: the lower frequency f_l , the central frequency f_c and the higher frequency f_h . On a Mel scale, the distances $f_c - f_l$ and $f_h - f_c$ are the same for each filter and are equal to the distance between the f_c 's of the successive filters.

Using the triangular filter bank, the spectral components are collected into bins. This scale uses smaller bins for lower frequencies, which are perceptually more important than higher frequencies. Figure 2.2 illustrates.

To implement the filterbank each windowed frame of speech data is transformed using a fast Fourier transform (FFT). The magnitudes of these coefficients are then binned by multiplication with each of the triangular filters. Binning means each FFT magnitude coefficient is multiplied by the corresponding filter gain and the results are accumulated. Therefore, each bin holds a weighted sum representing the spectral magnitude in that filter bank channel.

Normally, the triangular filters are spread over the whole frequency range from zero up to the Nyquist frequency. However, band limiting is often useful to reject unwanted frequencies or avoid considering frequencies in regions in which there is no useful signal energy. This is the case, for example, when processing telephone speech, which has no useful information above approximately 4kHz.

In order to compute the cepstra, the logarithm is taken of the filterbank energies (refer back to Figure 2.1) after which a lowpass filter is applied. This lowpass filter is normally implemented by applying a FFT and retaining the low frequency components. However a more efficient transform is applicable in this case: the Discrete Cosine transform (DCT).

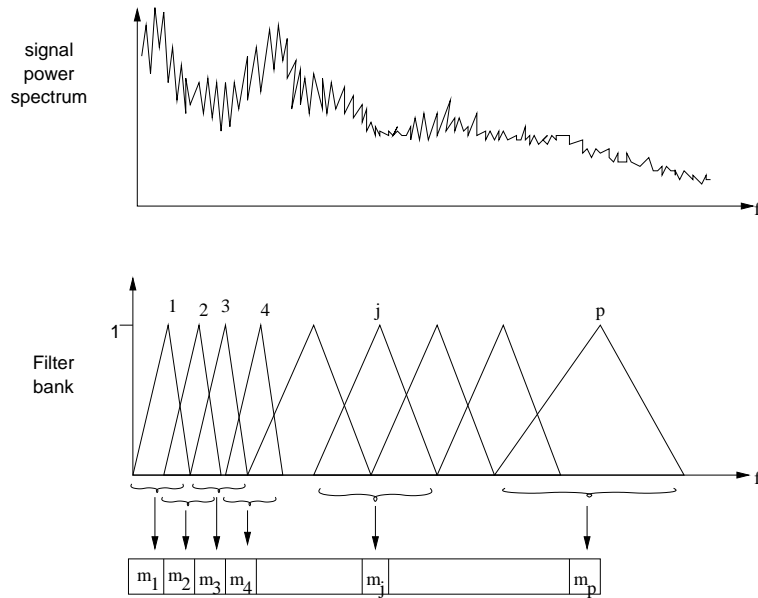


Figure 2.2: Triangular filter spread over a frequency spectrum according to Mel scale.

2.1.4 The Discrete Cosine Transform (DCT)

A number of methods can be used to obtain spectral transformations such as the Discrete Fourier Transform (DFT) and related FFT. However, the Discrete Cosine Transform is more efficient and more appropriate when the signal is real and even since it takes advantage of redundancies in the DFT. Since the filterbank amplitudes are real and even, the DCT can be used to derive cepstral coefficients from the Mel filterbanks. The following equation (2.5) shows how the cepstral coefficients are calculated using DCT:

$$c_i = \sqrt{\frac{N}{2}} \sum_{j=1}^N \log(m_j) \cdot \cos\left(\frac{\pi i}{N}(j - 0.5)\right), \quad (2.5)$$

where N corresponds to the number of filterbank channels, and $\log(m_j)$ to the log filterbank amplitudes.

Hence the coefficients that are obtained by applying the DCT to the log energies obtained from a Mel filterbank are termed MFCCs.

The various parameterisation used for language identification are described in the following sections.

2.1.5 MFCC_E, MFCC_E_D and MFCC_E_D_A

For language identification the lowest 13 coefficients of the Mel-cepstrum are calculated (c_0 through c_{12}), thereby retaining information relating to the speakers vocal tract shape while ignoring the excitation signal. This is the same approach often used by automatic speech recognition systems. The lowest cepstral coefficient (c_0) is replaced by the frame energy E . Due to the fact that coefficients in the Mel-cepstrum have a tendency not to be linearly related, they are considered to be a relatively orthogonal set [18].

The vector formed by the first 13 MFCC coefficients, but with the first C_0 replaced with the frame energy E , will be referred to in the remainder of this document as MFCC_E.

In an effort to model temporal transitions, a vector of cepstral difference can also be computed for every frame. These are sometimes referred to as the "delta" coefficients, given by

$$\Delta c_i(n) = c_i(n+1) - c_i(n-1) \quad (2.6)$$

Δc_0 is included as part of the delta-cepstral vector, thus making it a 13 coefficient vector.

The delta features of the n^{th} MFCC_E vector are computed as the difference between the $n^{\text{th}} + 1$ and the $n^{\text{th}} - 1$ vectors. This delta is appended to the n^{th} MFCC_E vector for the MFCC_E_D parameterisation. This process is depicted in figure 2.3.

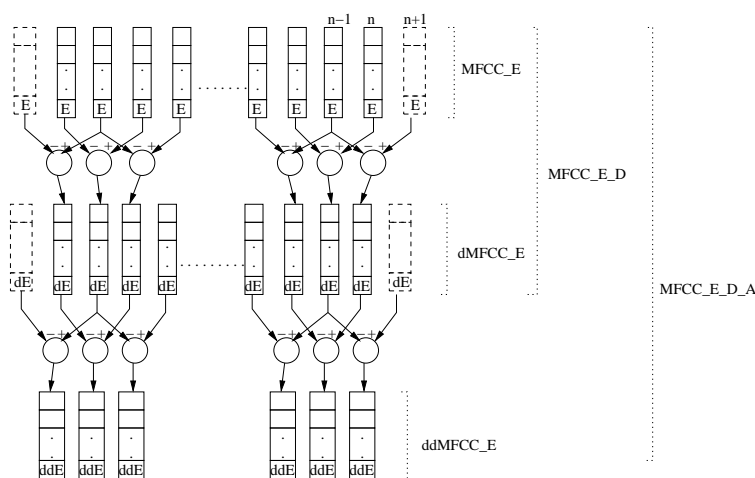


Figure 2.3: The MFCC feature vector extraction process.

Since the first vector in the frame has no predecessor, a phantom vector is assumed to exist, whose value is equal to that of the first vector. The difference between this phantom vector and the second vector is used to obtain the first dMFCC_E vector. The same procedure is used for the last vector in the frame.

These phantom vectors are indicated in Figure 2.3 by broken lines. A similar procedure is followed to obtain the second differential $ddMFCC_E$ frame from the first differential $dMFCC_E$ frame.

Figure 2.4 illustrates the different vector structures of an $MFCC_E$, $MFCC_dE$ and $MFCC_ddE$ feature vector.

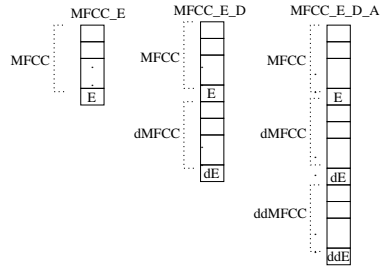


Figure 2.4: The different MFCC vector structures.

2.1.6 Shifted Delta Cepstra (SDC)

While MFCC feature vectors are typically formed by concatenating cepstra with their first and possibly also second differentials, SDC feature vectors are created by stacking delta cepstra computed across multiple speech frames. The computation of the SDC concatenates all the $\Delta c(t + iP)$ vectors,

$$\Delta c(t) = c(t + iP + d) - c(t + iP - d)$$

where

N is the number of cepstral coefficients computed at each frame.

d represents the time advance or time delay for the delta computation.

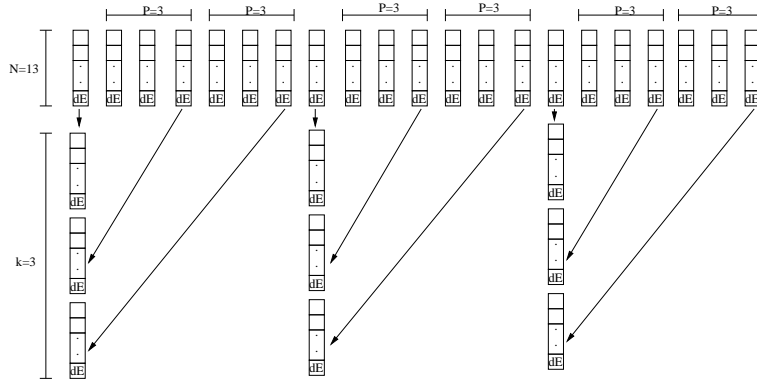
k is the number of blocks whose delta coefficients are concatenated to form the final feature vector.

P is the time shift between two consecutive blocks.

Shifted Delta Cepstra are computed using the first differential vectors $dMFCC_E$ as indicated in Figures 2.3 and 2.4. A total of k of these $dMFCC_E$ vectors are stacked to form the SDC vector, where each of the k $dMFCC_E$ is P frames from the previous one. This process is illustrated in Figure 2.5.

2.2 Statistical Models

Statistical models are important in LID systems, because they make it possible to classify a test utterance as belonging to one of the languages in a training set.

Figure 2.5: The SDC feature construction process for $k=3$ and $P=3$.

A statistical classification has the advantage of relying on the patterns found in training examples rather than hand-crafted rules regarding the features. Among the most widespread statistical models is the Gaussian distribution.

2.2.1 One-dimensional Gaussian Distribution

In one dimension (one feature), the Gaussian probability density function can be expressed as

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left[-\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2\right] \quad (2.7)$$

and its graphical representation is shown in figure 2.6.

The Gaussian density is considered to be one of the most important of all densities because of its accurate description of many real world quantities, especially when such quantities are the result of many small independent random effects acting to create the quantity of interest [9].

2.2.2 Two-Dimensional Gaussian Distribution

Two random variables x and y are said to be drawn from a Gaussian density function if it is of the form:

$$P(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \cdot \exp\left\{\frac{-1}{2(1-\rho^2)}\left[\frac{(x - \mu_x)^2}{\sigma_x^2} - \frac{2\rho(x - \mu_x)(y - \mu_y)}{\sigma_x\sigma_y} + \frac{(y - \mu_y)^2}{\sigma_y^2}\right]\right\} \quad (2.8)$$

This is sometimes called a bivariate Gaussian density function, a special case of the multivariate Gaussian density function.

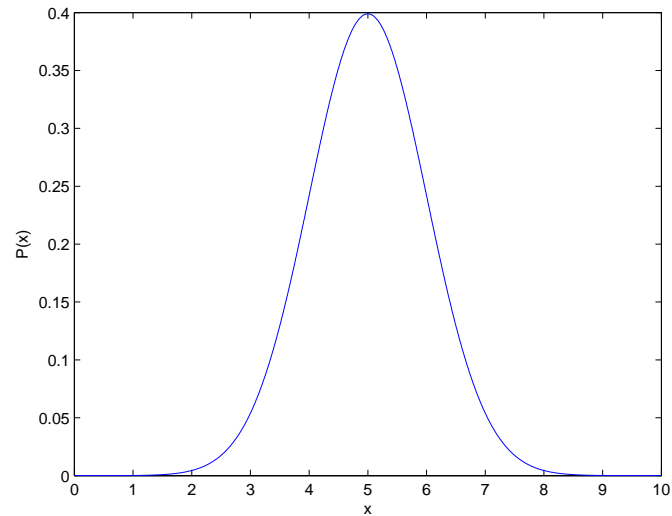


Figure 2.6: Unimodal Gaussian of a single random variable with $\sigma=1$ and $\mu=5$.

The parameters μ_x and μ_y are the means of the random variables x and y respectively, and σ_x and σ_y their standard deviations. The quantity ρ is known as the correlation coefficient and is given by

$$\rho = E[(x - \mu_x)(y - \mu_y)] / \sigma_x \sigma_y$$

In Figure 2.7 the bivariate density function is shown as scatter plot of the variables x and y .

2.2.3 N-Dimensional Gaussian Distribution

The multivariate Gaussian PDF of an $d \times 1$ random vector \mathbf{x} is defined as:

$$p(\mathbf{X}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \cdot \exp\left[-\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{X} - \boldsymbol{\mu})\right], \quad (2.9)$$

where $\boldsymbol{\mu}$ is the mean vector, $\boldsymbol{\Sigma}$ is the covariance matrix and $|\boldsymbol{\Sigma}|$ is the determinant of this matrix. $\boldsymbol{\Sigma}$ is assumed to be positive definite and thus $\boldsymbol{\Sigma}^{-1}$ exists. The covariance matrix is always symmetric about the diagonal, since $c_{ij} = c_{ji}$. The mean vector is defined as

$$[\boldsymbol{\mu}] = E(\mathbf{x})$$

where

$$\boldsymbol{\mu} = [\mu_1, \mu_2, \dots, \mu_d]$$

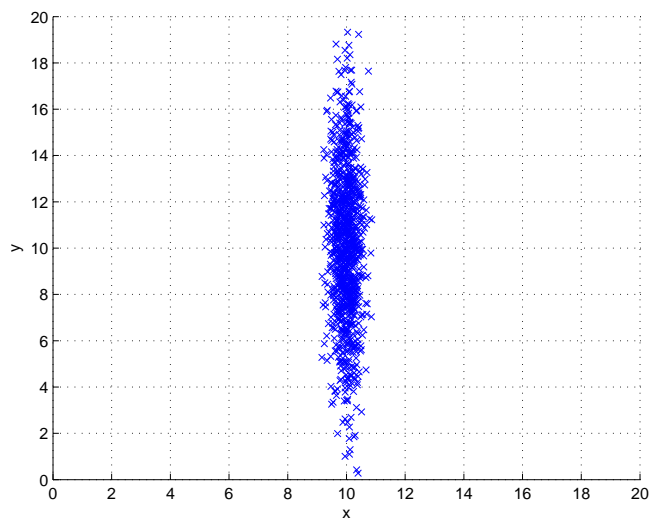


Figure 2.7: Scattergram of two random variables with $\mu_x = 10$, $\mu_y = 10$, $\sigma_x = 0.1$, $\sigma_y = 10$ and $\rho = 0$.

such that μ_i is the mean of random variable x_i . The elements of

$$\Sigma = \begin{bmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1d} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{d1} & \rho_{d2} & \cdots & \rho_{dd} \end{bmatrix},$$

which is called the *covariance matrix*¹, are given by

$$\rho_{ij} = E[(x_i - \mu_i)(x_j - \mu_j)] = \begin{cases} \sigma_{x_i}^2 & i = j \\ \sigma_{x_i}\sigma_{x_j} & i \neq j \end{cases} \quad (2.10)$$

2.2.4 Diagonal Covariance Approximation

If we assume that the off-diagonal elements of the covariance matrix Σ are zero, because the corresponding correlation coefficients ρ_{ij} with $i \neq j$ are null, we are left with only the diagonal elements.

Assuming a diagonal covariance is assuming *stastical independence*² between

¹A covariance matrix is merely a collection of many covariances in the form of a $d \times d$ matrix. The resulting covariance $C_{i,j}$ value will be larger than 0 if i and j tend to increase and decrease together, below 0 if they tend to increase and decrease in opposite directions, and 0 if they are independent.

²Two events are statistically independent, if the probability of their occurring jointly equals the product of their respective probabilities. When features x_i and x_j are statistically independent, their covariance is zero, i.e.; $\sigma_{ij}^2 = 0$.

the elements of the feature vector \mathbf{x} .

2.2.5 Maximum-likelihood parameter estimates for a Gaussian distribution

Assume we are given a set of data consisting of N feature vectors

$$\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^T.$$

Next we assume this data is Gaussian, and we would like to find the parameters of the Gaussian $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ that best describe the data.

The log likelihood function for the observed data \mathbf{x} is given by

$$L(\mathbf{x}) = \sum_{i=1}^N \log[P(\mathbf{x}_i)].$$

The Gaussian PDF of a random vector \mathbf{x}_i having a d -dimensional multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is denoted by Equation 2.9. Our aim is to find the parameters $\boldsymbol{\mu} = \boldsymbol{\mu}_{ML}$ and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{ML}$, which maximise the likelihood function $L(\mathbf{x})$. To find the optimum value for the mean we determine the derivative of the log likelihood function with respect to the mean, and we do like-wise for the covariance matrix [5].

The derivative of the log likelihood function with respect to the mean is given by

$$\frac{\partial L(\mathbf{x})}{\partial \boldsymbol{\mu}} = \sum_{n=1}^N \boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}) = 0 \quad (2.11)$$

From this it follows that the maximum likelihood estimate of the mean for a Gaussian distribution is the sample mean

$$\boldsymbol{\mu}_{ML} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n. \quad (2.12)$$

The derivative of the log likelihood function with respect to the covariance matrix is given by:

$$\frac{\partial L(\mathbf{x})}{\partial \boldsymbol{\Sigma}} = \frac{N}{2} \boldsymbol{\Sigma}^{-1} - \frac{1}{2} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Sigma}^{-1} = 0. \quad (2.13)$$

From this it follows that the maximum likelihood estimate of the covariance for a Gaussian distribution is the sample covariance

$$\boldsymbol{\Sigma}_{ML} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T. \quad (2.14)$$

2.2.6 Gaussian Mixture Models

A mixture model is a linear combination of M basis distribution given by

$$p(\mathbf{x}) = \sum_{j=1}^M \alpha_j \cdot P_j(\mathbf{x}), \quad (2.15)$$

where

- $P(\mathbf{x})$ is the j^{th} basis distribution, which is assumed to be Gaussian for a Gaussian Mixture Model (GMM), and
- α_j is the j^{th} mixture weight with $0 \leq \alpha_j \leq 1$ and $\sum_{j=1}^M \alpha_j = 1$.

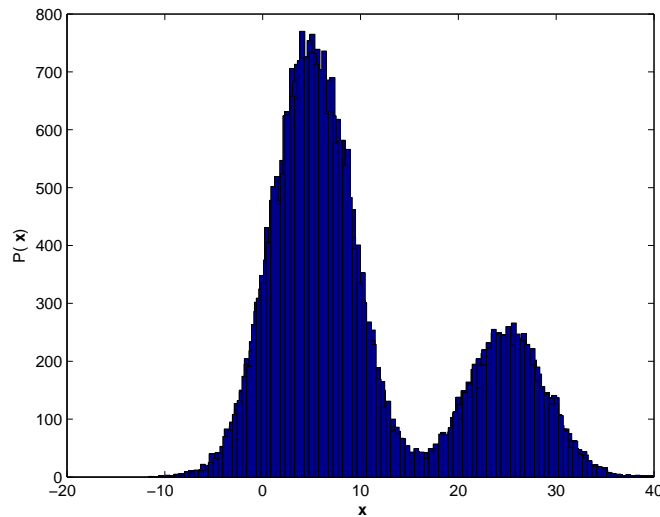


Figure 2.8: Bimodal Gaussian Histogram.

A mixture model is able to represent a wider variety of distributions than the single Gaussian, such as multimodal, non-symmetric and correlated distributions when using diagonal covariances. However, it is now more difficult to determine the parameters of the individual mixtures, and the mixture weights, for a set of data. The EM algorithm has been derived for this purpose.

2.2.7 EM Algorithm

The EM Algorithm is an iterative optimization of the means, variances and mixture weights of the M basis distributions of a Gaussian mixture model. The aim is to optimize the likelihood that the given data points are generated by

the mixture of Gaussians [1]. The EM algorithm alternates between performing an expectation (E) step, and a maximisation (M) step.

- E - computes an expectation of the likelihood by including the *latent variables*³ as if they were observed variables.
- M - estimates the parameters by maximising the expected likelihood found in the E step.

This technique is commonly referred to as the *Expectation Maximisation* (EM) *algorithm*. The main idea of EM is to estimate the densities by taking an expectation of the logarithm of the joint density between the known and the unknown components, and then maximise this function by updating the parameters that are used in the probability density function. In order to find the updated parameters (i.e., means, variances and mixture weights) that give a good representation of the true distribution, the parameters must be updated iteratively using the EM algorithm until the expected likelihood converges to a stable value, indicating that an optimum has been reached.

The process begins by assigning a set of initial values for the unknown parameters (e.g., $\boldsymbol{\mu}$ means of mixtures must differ on initialisation, $\sigma^2 = 1$ and $\boldsymbol{\Sigma} = I$, the identity matrix, and the mixture weights $\alpha_i=1/M$). The training process continues until the likelihood reaches a locally optimal value.

The basic function used in the training process take the form of a Gaussian distribution, in which each base function is represented by a mean $\boldsymbol{\mu}$, variance σ^2 and a mixture weight $\alpha(i)$. The update equations of the EM algorithm for the parameters of this distribution are the following

$$\mu_j^{new} = \frac{\sum_n \gamma_j^{old}(x^n) x^n}{\sum_n \gamma_j^{old}(x^n)} \quad (2.16)$$

$$(\sigma_j^{new})^2 = \frac{1}{d} \frac{\sum_n \gamma_j^{old}(x^n) \|x^n - \mu_j^{new}\|^2}{\sum_n \gamma_j^{old}(x^n)} \quad (2.17)$$

$$\alpha_j^{new} = \frac{1}{N} \sum_n \gamma_j^{old}(x^n) \quad (2.18)$$

where

$$\gamma_j(x) = \frac{p_j(x)\alpha_j}{\sum_{j=1}^M p_j(x)\alpha_j} \quad (2.19)$$

³Latent variables are variables that are not directly observed, but are rather inferred from other variables that are observed and directly measured. In the case of a GMM, the identity of the mixture from which a data point is drawn is such a latent variable.

2.2.8 Hidden Markov Models

An HMM is a stochastic finite state process where each state has an associated observation probability distribution which determines the probability of generating the observation \mathbf{o} at time t . Only one state of an HMM is occupied at any given time, and the occupation moves from one state to the next at discrete time intervals. The cost of moving to the next state is determined by the transitional probability a_{ij} which is associated with each pair of states. The probability of transiting from one state to another is dependent only on the current state and not on any previous states. Stated mathematically

$$\begin{aligned} P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \dots) \\ = P(q_t = S_j | q_{t-1} = S_i) \end{aligned} \quad (2.20)$$

This equation states that if the state occupied at time $t-1$ was S_i , then the state occupied before $t-1$ such as S_k becomes irrelevant with respect to the probability of a transition from state S_i to S_j [10]. The transition probability from the current state i to the next state j is usually written as $a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$. Hence the transition probabilities within an N state HMM can be written as an $N \times N$ matrix. This implies that the model dependencies between adjacent observations are captured by stochastic dependencies between the hidden states. Sometimes an additional but non-emitting pair of entry and exit states are also included. This facilitates the later interconnection of several HMMs into a larger network.

Each state of the HMM has an output probability distribution which determines the output of the HMM when it is in a given state. The output probability distribution of the HMM is sometimes referred to as the emission probabilities of the HMM. The parameters of the HMM are determined from training observation sequences using a form of EM algorithm, known as the Baum-Welch algorithm [11]. The Viterbi algorithm [11] is used for classifying an input vector sequence with a given HMM. However, the Viterbi algorithm may also be used to estimate the HMM parameters.

In constructing an HMM the first step is to choose a *priori* a topology for each HMM. This topology consists of:

- The number of states.
- The form of the observation probability density function that is associated with each state.
- The arrangement of transitions between states.

The model structure we will use later in this thesis consists of one active state s_2 , while s_1 and s_3 are non emitting states, and have no associated observation probability density. The observation function b_2 is a Gaussian mixture model with diagonal or full covariance matrices. Figure 2.9 is a diagrammatic representation of this single state HMM.

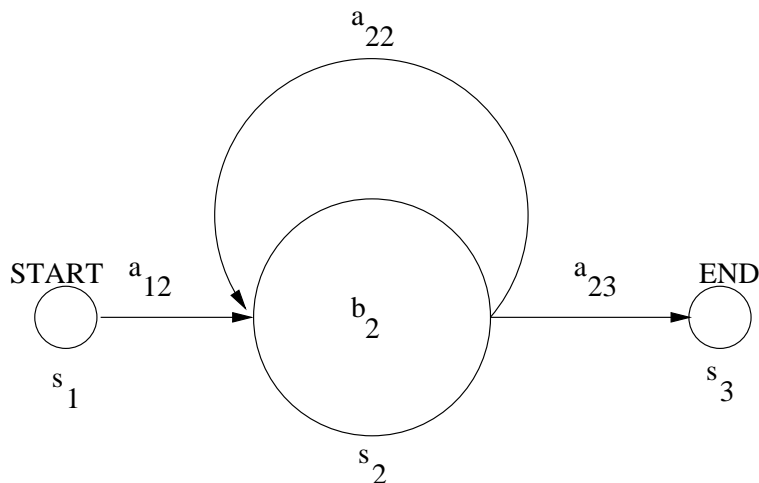


Figure 2.9: Illustration of a single state HMM.

2.2.9 Training HMMs

In maximum likelihood estimation we try to maximise the likelihood of a given sequence of observations \mathbf{O} , given the HMM λ , expressed mathematically as

$$L = P\{\mathbf{O}|\lambda\}.$$

There is no known way to analytically solve for the model $\lambda = (A, B, \pi)$, which maximise the quantity $L = P\{\mathbf{O}|\lambda\}$. But we can choose model parameters such that it is locally maximised, using an iterative procedure, which is described below.

We have a model λ and a sequence of observations $\mathbf{O} = \mathbf{o}_1, \mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_T$, and $P(\mathbf{O}|\lambda)$ must be found. We can calculate this quantity using simple probabilistic arguments by considering each possible way the observation sequence can be generated by the HMM. However, this calculation involves a number of operations in the order of N^T . This is very large even if the length of the sequence, T is moderate. Therefore we have to look for another method for this calculation. Fortunately there exists one which has a considerably lower complexity and makes use of an auxiliary variable, $\alpha_t(i)$ called the forward variable.

The forward variable is defined as the probability of the partial observation sequence $\mathbf{o}_1, \mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_t$, when it terminates at the state i at time t . Mathematically, we can express this as

$$\alpha_t(i) = P(\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_t, q_t = i|\lambda).$$

It can then be shown that the following recursive relationship holds:

$$\alpha_{t+1}(j) = b_j(\mathbf{o}_{t+1}) \sum_{i=1}^N \alpha_t(i) a_{ij}, \quad (2.21)$$

where

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1) \quad 1 \leq i \leq N$$

and π_j is the probability of the sequence beginning in state j . From the definition of $\alpha_t(i)$ it then follows that:

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

In a similar way we can define the backward variable, $\beta_t(i)$ as the probability of the partial observation sequence $\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, \dots, \mathbf{o}_T$, given that the current state at time t is i . Mathematically, we can write:

$$\beta_t(i) = \sum_{j=1}^N \beta_{t+1}(j) a_{ij} b_j(\mathbf{o}_{t+1}) \quad 1 \leq i \leq N, \quad 1 \leq t \leq T-1 \quad (2.22)$$

where, the recursion begins with:

$$\beta_T(i) = 1, \quad 1 \leq i \leq N$$

From the definition of the forward and backward variables it can be shown that:

$$P(\mathbf{O}|\lambda) = \alpha_N(T) = \beta_1(T)$$

Further it follows that,

$$\alpha_t(i) \beta_t(i) = P(\mathbf{O}, q_t = i | \lambda) \quad 1 \leq i \leq N, \quad 1 \leq t \leq T-1$$

Therefore this gives another way to calculate $P(\mathbf{O}|\lambda)$, by using both forward and backward variables as follows:

$$P\{\mathbf{O}|\lambda\} = \sum_{i=1}^N P(\mathbf{O}, q_t = i | \lambda) = \sum_{i=1}^N \alpha_t(i) \beta_t(i)$$

The calculation of $P(\mathbf{O}|\lambda)$ as indicated above is known as the forward-backward procedure. The Baum-Welch algorithm can be described in terms of the forward-backward procedure [11]. To do this, we use the forward and backward probabilities to write down the probability of being in state i at time t and in state j at time $t+1$:

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j | \mathbf{O}, \lambda) \quad (2.23)$$

Using Bayes rule, this can be expressed as:

$$\xi_t(i, j) = \frac{P(q_t = i, q_{t+1} = j, \mathbf{O}|\lambda)}{P(\mathbf{O}|\lambda)}.$$

Using forward and backward variables this can be expressed as:

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(\mathbf{O}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)a_{ij}b_j(\mathbf{O}_{t+1})\beta_{t+1}(j)}. \quad (2.24)$$

This leads to the following expression for the updated transition probabilities:

$$\bar{a}_{ij} = \frac{\sum_{t=1}^T \xi_t(i, j)}{\sum_{t=1}^T \sum_{j=1}^N \xi_t(i, j)} \quad (2.25)$$

In this equation the numerator corresponds to the expected number of transitions from state i to state j .

A similar approach can be taken to derive update equations for the means and variances of Gaussian probability distributions at state j . First we obtain an equation for the probability of occupying state j at time t :

$$\begin{aligned} L_j(t) &= P(q_t = j|\mathbf{O}, \lambda) \\ &= \frac{P(q_t = j, \mathbf{O}|\lambda)}{\sum_{k=1}^N P(q_t = k, \mathbf{O}|\lambda)} \\ &= \frac{\alpha_j(t)\beta_j(t)}{\sum_{k=1}^N \alpha_k(t)\beta_k(t)}. \end{aligned} \quad (2.26)$$

Then the updated means are given by:

$$\bar{\mu}_j = \frac{\sum_{t=1}^T L_j(t)\mathbf{o}_t}{\sum_{t=1}^T L_j(t)} \quad (2.27)$$

and the variance by:

$$\bar{\Sigma}_j = \frac{\sum_{t=1}^T L_j(t)(\mathbf{o}_t - \bar{\mu}_j)(\mathbf{o}_t - \bar{\mu}_j)^T}{\sum_{t=1}^T L_j(t)}. \quad (2.28)$$

In each case the numerator weights the observations with the probability of occupation at each time t of the respective state j .

Equations (2.25), (2.27) and (2.28) can be used to update the parameters of an HMM with Gaussian emission probability density functions, and are known as the Baum-Welch equations. Once the transition probabilities, Gaussian means and Gaussian covariances have been updated, the forward and backward variables must be recalculated, after which the parameters can be updated again. This iterative procedure is usually carried out until the probability of the data $P(\mathbf{O}|\lambda)$ converges.

The procedure described above can also easily be extended to Gaussian mixture emission distributions by representing each mixture as an HMM with parallel single mixture states and transition probabilities corresponding to the mixture weights. A similar transformation will be applied in our experiments in Section 5.6.

2.3 Summary

In this chapter the calculation of MFCC and SDC features from the speech signal were reviewed. Statistical modelling techniques that make it possible to model these feature vectors by Gaussian distributions were then discussed. These Gaussians provide a general indication as to how the features of the signal are distributed.

The EM algorithm which is used to obtain the parameters of a GMM given a set of training vectors was reviewed. Finally HMMs which are able to model sequences of feature vectors were described, and the Baum-Welch algorithm which is used to train them was introduced.

Chapter 3

A Survey Of Multi-Lingual Speech Corpora

In this chapter, the data corpora that have been used by other researchers for the development of language identification systems will be reviewed and compared. Finally, the corpus that has been compiled for our systems will be described and experiments will be introduced.

3.1 OGLTS

The Oregon Graduate Institute Multi-lingual Telephone Speech Corpus (OGLTS) is a speech corpus conceived for the purpose of conducting research on automatic language identification. In 1992 the corpus consisted of the 10 languages listed in Table 3.1 [7]. In 1994 the corpus was extended by the addition of Hindi, which brought the total number of languages to 11 [6].

Language	No. of Speakers	Duration (hrs)
English	299	7.22
Farsi	153	3.23
French	149	3.23
German	157	3.49
Japanese	147	3.20
Korean	148	2.55
Mandarin	174	3.12
Spanish	149	3.33
Tamil	188	3.23
Vitnamese	158	3.03
Total	1722	37.41h

Table 3.1: Composition of the OGLTS corpus.

3.1.1 Selection Of Languages

In selecting the languages, several factors were taken into account. Firstly, the availability of native speakers in the United States was considered. Secondly, known relationships and differences that exist between the selected languages played a role. For example, English and German are of Germanic origin, while French and Spanish are of Latin origin. Linguistic characteristics such as the use of pitch and accents in Japanese as opposed to tonal languages like Mandarin Chinese and Vietnamese also formed a basis for consideration. Finally, the selected languages represent important geographic and political regions.

3.1.2 Data Collection Process

The data was collected as a campaign under the theme "donate your voice to science", in which speakers volunteered to participate in the research project. An interactive graphical interface played excerpts of speech at random in each of the 10 languages, prompting listeners to respond. A log was maintained of all the responses. Initially, callers received a greeting in English followed by a prompt to select a language by means of the digits 0 to 9. Thereafter, the prompts were given in the target language only. The recordings included fixed vocabulary items, short topic specific descriptions and samples of elicited free speech, which callers were prompted to utter after having been given an opportunity to prepare themselves for the actual recording. Examples of the prompts and typical responses are:

1. Prompts for obtaining fixed vocabulary.

Q: What is your native language?

A: Japanese.

Q: Please say the numbers zero through ten.

A: zero, one, two, three, four, five, six, seven, eight, nine, ten.

2. Prompts for obtaining topic specific descriptions.

Q: Describe the room that you are calling from.

A: The room is small, it has a window and the wall is painted white.

Q: Describe your most recent meal.

A: I had a cheese burger with lettuce and tomato.

3. Prompts for obtaining free speech [8].

Q: We want you to talk for a longer period, we do not care what you say. You have 1 minute to say it, and we will give you 10 seconds to think about it. Please do not read.

All speech was sampled at 8000 samples/sec at 14 bit resolution.

3.1.3 Corpus Validation And Annotation

The corpus was put through a preliminary screening phase, in which the recordings were edited for excess noise and/or silences. Thereafter, broad phonetic transcriptions were compiled. The phonetic categories used were vowels, fricatives, stops, silences or background noise, and vocalic sonorants.

A subsequent control phase followed, in which the broad phonetic transcriptions were verified by a native speaker of the individual language. Furthermore, detailed phonetic transcriptions were produced for small portions of the data, as well as time aligned syllable boundaries. Orthographic transcriptions were also compiled for each language by native speakers [7].

3.2 GLOBALPHONE

GlobalPhone is a database of high quality read speech and text data in a variety of languages, which is suitable for the development of large vocabulary speech recognition systems [12]. It covers the 15 languages listed in Table 3.2. The corpus contains more than 300 hours of transcribed speech by more than 1500 native adult speakers.

Language	No. of Speakers	Duration (hrs)
Arabic	170	35
Ch-Mandarin	132	31
Ch-Shanghai	41	10
Croatian	92	16
Czech	102	29
French	94	25
German	77	18
Japanese	144	34
Korean	100	21
Portuguese	101	26
Russian	106	22
Spanish	100	22
Swedish	98	22
Tamil	49	N/A
Turkish	100	17
Total	1506	328h

Table 3.2: Composition of the GlobalPhone corpus.

With the aim of deploying a Large Vocabulary Continuous Speech Recognition (LVCSR) system, an average of 20 hours of transcribed speech was collected per language. The domain chosen for GlobalPhone made it possible to collect suitable large text corpora from the web.

3.2.1 Selection Of Languages

Given that it is estimated that there are more than 4 500 languages in the world and only 150 of these are spoken by over a million people, the following characteristics were considered for selecting the representative subset of languages:

1. The size of the speaker population.
2. Political and economic relevance.
3. Geographic coverage.
4. Phonetic coverage.
5. Orthographic speech variety, for example, alphabetic speech like Latin, syllable based languages like Japanese, and ideographic texts like Chinese.
6. Morphologic variety, such as agglunative languages like Turkish.

While the GlobalPhone languages were selected following these criteria, equal importance was not given to each. For example, the size of the speaker population was favoured over geographic coverage, hence no African language was selected.

Considering that the most time-consuming process in the compilation of a speech database is the transcription, GlobalPhone collected speech data read from text that was already electronically available. For this purpose widely read newspapers available on the internet were selected as resources, and text from national and international political and economic topics were chosen to restrict the vocabulary.

All GlobalPhone data was collected in the home countries of the native speakers. This was done to avoid the inclusion of unavoidable artifacts associated with collecting speech of speakers living in non-native environments, for instance, a native Brazilian living in Portugal.

3.2.2 Data Collection Process

In the acquisition process GlobalPhone recorded approximately 100 native speakers per language, with each speaker session lasting approximately 20 minutes. The speakers were allowed to familiarise themselves with the prompting text before recording in order to clarify pronunciations and minimise reading errors.

Most of the recordings were done in small quiet rooms, with the exception of a few recordings done in public, but quiet environments.

Recordings were made using a portable Sony TDC-8 DAT recorder and a close talking Sennheiser HD-440-6 microphone. The data was recorded at a 48-KHz sampling rate and 16-bit linear quantisation, and subsequently down-sampled for further processing.

3.2.3 Corpus Validation and Annotation

The recorded data was validated in a two-step process. First, an automatic silence detector split the files into sentences. Second, human listeners checked if the speech corresponds to the prompting text. Incorrectly read utterances with major differences to the prompts were deleted from the database.

In order to control the data proportions, demographic information from each speaker was collected including, gender, age, up-bringing, level of education and state of health (such as colds or allergies).

For each language the data was then divided into three sets: one set for training (80%), one set for cross validation (10%) and one for evaluation (10%). No speaker appears in more than one set and no article is read more than once.

3.3 CALLFRIEND

From 1993 to 1996 the National Institute of Standards and Technology (NIST) of the United States Defence Department has sponsored evaluation of language identification systems using the OGLTS corpus. However, in 1996 the NIST evaluations adopted the Linguistic Data Consortium's CALLFRIEND corpus for further work.

The major difference between OGLTS and CALLFRIEND is that, while the former consisted mostly of read speech, the latter consists exclusively of unprompted conversational speech.

3.3.1 Selection Of Languages

The CALLFRIEND corpus was designed to consist of the same 11 languages that had been used in the OGLTS corpus. In 1996 Arabic was added to the 11 languages bringing the number of languages to 12, as listed in Table 3.3 [19].

3.3.2 Data Collection Process

The speech segments in the CALLRIEND corpus are all telephone conversational data, with each segment limited to one side of the conversation, and ranging from 5 to 30 minutes in length. It is presented sampled data in standard 8-KHz μ -law [2].

3.3.3 Validation and Annotation

The majority of the calls in the CALLFRIEND corpus have not been transcribed. An exception to this are 120 30-minute calls in Spanish and Mandarin Chinese [4]. As a result this corpus has not undergone a validation process as used in the compilation of the OGLTS and GLOBALPHONE corpora.

Language	No. of Calls	Duration (min)
Arabic	60	5-30
Farsi	60	5-30
German	60	5-30
Japanese	60	5-30
Korean	60	5-30
Tamil	60	5-30
Vietnamese	60	5-30
Mandarin	120	10-60
English	120	10-60
Hindi	60	5-30
Spanish	120	10-60
French	60	5-30
Total	900	approx. 75-450min

Table 3.3: Composition of CALLFRIEND corpus.

3.4 The Sub-Saharan Language Corpus

The Sub-Saharan Language Corpus (SSLC) is a telephone speech corpus compiled for the purpose of this research. It consists of 21 languages spoken in the southern part of Africa, as listed in Table 3.5. It includes several languages with European origins, for example, Portuguese, English, German and Russian. It also includes Arabic and some languages originating from Asia, but that are commonly spoken in the Sub-Saharan region. All speech in the corpus is spontaneous and unprompted.

3.4.1 Selection Of Languages

The languages were chosen opportunistically by virtue of their frequent occurrence in South Africa's mobile and fixed telephone networks. Relationships between languages or their phonetic characteristics were not taken into account explicitly. Rather, those languages for which at least 40 telephone conversations with a total duration of at least 60 minutes were selected for inclusion in the corpus. The following gives a brief description of the origins and usage for each language listed in Table 3.5.

- Afrikaans is a west-Germanic language spoken in South Africa. It is a variant of Dutch with some lexical and syntactic borrowing from Malay, Bantu, Khoisan, Portuguese and other European languages. In North America it is spoken in Canada and the United States. In Oceania it is spoken in Australia and New Zealand. In Africa it is also spoken in Lesotho, Malawi, Namibia, Swaziland, Zambia and Zimbabwe.
- Arabic is a Semitic macrolanguage of Saudi Arabia, spoken in at least 30 countries with each country speaking its own variant. In many in-

stances a country may even have more than one variant of the language. The following Arabic dialects can be distinguished: Saharan (Algeria), Algerian (Algeria), Babalia Creole (Chad), Baharna (Bahrain), Chadian (Chad), Cypriot (Cyprus), Dhofari (Oman), Bedawi (Egypt), Egyptian (Egypt), Gulf (Iraq), Hadrami (Yemen), Hijazi (Saudi Arabia), Libyan (Libya), Moroccan (Morocco), Najdi (Saudi Arabia), North Levantine (Syria), Mesopotamian (Iraq), Omani (Oman), Saidi (Egypt), Sanaani (Yemen), Shihhi (United Arab Emirates), South Levantine (Jordan), Standard Arabic (Saudi Arabia), Sudanese Creole (Sudan), Sudanese (Sudan), Taizzi-Adeni (Yemen), Tajiki (Tajikistan), Tunisian and Uzbeki (Uzbekistan).

- Chichewa is the alternate name for Nyanja. It is a southern Bantu language of Malawi. It is also spoken in Botswana, Mozambique, Swaziland, Zambia and Zimbabwe.
- English is a west-Germanic language of the United Kingdom. It is however widely used outside the U.K., and spoken in more than 110 countries, 28 of which are African. It is particularly prevalent throughout Southern Africa. However this thesis will focus on the varieties spoken in South Africa (South African English across all mother-tongues).
- German is west-Germanic language of Germany. It is widely used throughout Europe and Russia, and to a lesser extent in South America. In Africa it is spoken in Mozambique, Namibia and South Africa.
- Gujarati is an Indo-Aryan language of India. It is not widely used in Europe outside the U.K., but can be heard in the U.S.A. and Canada. It is also used in the Asian countries of Bangladesh, Indonesia and Singapore, and in the Middle-Eastern countries of Oman and Pakistan. In Africa it is spoken in Botswana, Kenya, Malawi, Mauritius, Mozambique, Reunion, South Africa, Tanzania, Uganda, Zambia and Zimbabwe.
- Hindi is an Indo-Aryan language of India. In Europe it is spoken in Germany and the United Kingdom. In North America it is spoken in Canada and United States. In Asia it is spoken in Bangladesh, Bhutan, Nepal, Philippines and Singapore. In the Middle-East it is spoken east in the United Arab Emirates and Yemen. In Africa it is spoken in Botswana, Djibouti, Kenya, South Africa, Uganda and Zambia.
- Kinyarwandi is an alternate name for Rundi. It is a southern Bantu language of Rwanda. It is also used in Burundi, the Democratic Republic of the Congo and Uganda.
- Kirundi is an alternate name for Rundi. It is a southern Bantu language of Burundi. It is also spoken in Rwanda, Tanzania and Uganda.
- Lingala is a southern Bantu language of the Democratic Republic of Congo. It is also spoken in Central African Republic and Congo.

- Luganda is an alternate name for Ganda. It is a southern Bantu language of Uganda, but also spoken in Tanzania.
- Nigerian is a macrolanguage which refers to a group of 527 languages spoken in Nigeria. However the official languages belonging to this macrolanguage are Edo, Efik, Adamawa Fulfulde, Hausa, Idoma, Igbo, Central Kanuri and Yoruba.
- Portuguese (Angola and Mozambique) is a latin language of Portugal. It can be heard in other European countries, including France and Spain. It is also widely used through out South America. In Africa it is spoken in Angola, Cape Verde Islands, Congo, Guinea-Bissau, Malawi, Mozambique, Senegal, South Africa and Zambia.
- Russian is a Slavic language of the Russian Federation. It is widely used in East-European countries, and can be heard in Canada and the U.S.A.. In Africa it is spoken in Mozambique.
- Shangaan is an alternate name for Tsonga. It is a southern Bantu language of South Africa. It is also spoken widely in Mozambique, Swaziland and Zimbabwe.
- Shona (Zimbabwe) is southern Bantu language of Zimbabwe. It is also spoken in Botswana, Malawi, South Africa and Zambia.
- Sotho (Southern) is a southern Bantu language of Lesotho. It is also spoken widely in Botswana, South Africa and Swaziland.
- Swahili (DRC) It is a southern Bantu language of the Democratic Republic of Congo.
- Swahili (Tanzania) is a southern Bantu language of Tanzania. It can also be heard in the U.S.A. and Canada, as well as the Middle-Eastern countries Oman and the United Arab Emirates. In Africa it is also spoken in Burundi, Kenya, Libya, Mayotte, Mozambique, Rwanda, Somalia, South Africa and Uganda.
- Urdu is an Indo-Aryan language of Pakistan. In Europe it is spoken in Germany, Norway and the United Kingdom, while in North America it is used in Canada and the United States. In the Middle East it is spoken in Afghanistan, Bahrain, Oman, Qatar, Saudi Arabia and the United Arab Emirates, and can also be heard in Bangladesh, India, Nepal and Thailand. In Africa it is spoken in Botswana, Malawi, Mauritius, South Africa and Zambia.

Most of the languages in the SSLC corpus are therefore Southern Bantu languages, followed by Germanic and Indo-Aryan, as indicated in Table 3.4.

Language family	No. of occurrences
Germanic	3
Latin	2
Slavic	1
Indo-Aryan	3
Semetic	1
Southern Bantu	11

Table 3.4: Frequency of occurrence of various language families in the SSLC corpus.

Language	No. of files	Total length (hrs)	Average length (min.)	Standard deviation (min.)
Afrikaans	140	11.95	5	4
Arabic	120	9.79	4	3
Chichewa	172	12.84	4	3
English	106	7.25	4	3
German	46	9.68	12	10
Gujarati	78	3.61	2	2
Hindi	120	10.23	5	5
Kinyarwanda	58	3.96	4	3
Kirundi	60	5.30	5	4
Lingala	112	6.44	3	3
Luganda	78	4.08	3	2
Nigerian	120	5.55	2	2
Portuguese (Ang)	124	7.67	3	2
Portuguese (Moz)	134	6.05	2	1
Russian	76	7.55	5	4
Shangaan	106	3.45	1	1
ShonaZim	158	14.56	5	5
Sotho	126	6.41	3	2
Swahili (DRC)	136	6.12	2	2
Swahili (Tza)	120	8.86	4	4
Urdu	138	9.72	4	3
Total	2386	164.62h	4.11	3.2

Table 3.5: Composition of the SSLC corpus before data preparation.

3.4.2 Data Collection Process

The raw data is encoded as 8-kHz stereo A-law, with one conversation side per stereo channel. A number of processing steps were applied to this raw data before it was used in experimental evaluations, and these will be described in Section 3.5.

3.4.3 Data Evaluation

The raw data were evaluated by qualified language specialists who are acquainted with the language in the corpus. For each speech file, only the identity of the language was determined. No orthographic or phonetic transcription was performed.

3.5 Data Preparation for the Sub-Saharan Language Corpus

The raw data was obtained on CD as A-law encoded Microsoft WAV files with a sample rate of 8 kHz. The recordings are stereo, with one channel for each side of the telephone conversation. The following sections describe the processing applied to this data prior to its use in the LID system.

3.5.1 Naming convention

A uniform file naming convention was adopted, with each stereo WAV file given a name beginning with the language in question, followed by a suffix to differentiate different files of the same language. For example

lingala_1166sec.wav

indicates a file lasting 1166 seconds in the lingala corpus.

3.5.2 File format conversion

The source WAV files were converted to 16-bit linear PCM NIST SPHERE format for ease of subsequent processing by the HTK tools. This conversion was achieved using an open- source software tool called SoX (Sound Exchange).

Furthermore, SoX was used to split left and right channels into individual files, containing the separate sides and therefore the separate speakers of each conversation. For example, the stereo file

lingala_1166secR.12.sph

would be split into two files

lingala_1166secL.sph

and

lingala_1166secR.sph

3.5.3 Silence pruning and speech file segmentation

Significant portions of the separated left and right channels of the conversation were taken up by silence. This is not useful information, and must be discarded.

Furthermore, *Cross talk*¹, other speakers in the environment, and the telephone handset used can contribute to noise. All these factors pose some challenges for the purpose of eliminating the silence segments in an audio file. In order as far as possible to use only meaningful speech data for further processing, the silent portions have to be pruned from the audio file.

An in-house developed tool was used to remove silences from the audio files by partitioning the files into smaller segments. It does this by establishing an energy threshold that is considered to be the lowest energy level that speech is considered to have. Any segments of the audio file whose energy level is below this threshold are considered to be silence, and therefore discarded. The energy is calculated per frame using equation 3.1.

$$E = \left| \sum_{n=0}^{N-1} x(n) \right|^2 \quad (3.1)$$

Where $x(0) \dots x(N-1)$ are the N samples of a speech frame. The minimum number of frames that can constitute a speech segment is set at 32 frames, frames are composed of 256 samples each. This avoids impractically small portions of speech to be considered as individual segments. A speech segment must be encapsulated by 10 silence frames at the beginning and end. These smaller speech fragments were found to be not longer than 1 minutes at the most, and were saved in files and their names are appended with an ascending number index so that they can be distinguished. For example, the left channel file

lingala_1166secL.sph

may be split into a number of speech segments, each of which is named

lingala_1166secL.1.sph

lingala_1166secL.2.sph

etc.

By listening to a sample of the resulting files it was verified that this process does a good job of eliminating the silence, although is not robust enough to eliminate noise present within silent segments. The pruning of silences substantially reduces the length of the remaining audio data.

3.5.4 Set division

At this point, the database consists of audio files of the various languages in varying lengths. In an attempt to adhere to the norm of speech data distribution

¹The voice of the left channel audible on the right and vice-versa.

Language	Development set		Evaluation set		Training set	
	No. of files	Total length (h)	No. of files	Total length (h)	No. of files	Total length (h)
Afrikaans	567	0.85	812	1.21	3180	4.49
Arabic	311	0.40	1099	1.23	2941	3.41
Chichewa	234	0.28	713	0.79	3962	4.62
English	159	0.21	396	0.64	1797	3.06
German	271	0.38	554	1.05	2045	3.64
Gujarati	109	0.14	259	0.29	1146	1.36
Hindi	139	0.20	427	0.64	3379	4.70
Kinyarwanda	288	0.50	178	0.25	1138	1.59
Kirundi	281	0.37	133	0.21	1537	2.12
Lingala	307	0.55	442	0.64	1307	2.02
Luganda	129	0.12	211	0.23	1372	1.71
Nigerian	128	0.15	401	0.46	1481	1.91
Portuguese (Ang)	143	0.19	648	1.09	2018	2.78
Portuguese (Moz)	172	0.20	456	0.61	1547	2.29
Russian	188	0.45	255	0.44	1823	3.23
Shangaan	140	0.14	282	0.29	959	1.04
ShonaZim	282	0.33	987	1.55	3960	5.11
Sotho	295	0.34	467	0.50	1701	2.10
Swahili (DRC)	144	0.21	503	0.63	1481	2.03
Swahili (Tza)	151	0.18	453	0.51	2699	3.38
Urdu	165	0.25	570	0.69	2960	3.93
Total	4603	6.440	10246	13.95	44433	60.52

Table 3.6: File distribution in the SSLC corpus after data preparation.

in a corpus, we divided our data into three sets: the development test set, the evaluation test set and the training test set. These were taken from the each language in the approximate proportions 10:10:80 for development, evaluation and training sets, respectively.

Prior to dividing the corpus into data sets the length of the audio files had to be established, in order to use the shorter files for testing and the longer files for training. The details pertaining to the files distribution in the database is displayed in Table 3.6.

The purpose of the development test set was to tune LID system parameters, whilst that of the evaluation is to test the performance of the system. The purpose of the training set is to obtain (train) the statistical models. Most (80%) of the data is reserved for training since a larger training set usually leads to improved system performance. Table 3.6 shows the final distribution of languages used in our corpus.

3.6 Summary

Most of the corpora discussed in this chapter are composed of European and Asian languages, and were recorded in laboratory conditions that are less prone to environmental noise. Often speech was prescribed, although in some specific cases efforts were made to record free speech.

In contrast, our corpus is composed entirely of free speech that is prone to environmental noise. The languages are predominantly African but also includes a few languages of European and Asian origin.

Chapter 4

GMM LID Systems

This chapter is a literature review of LID systems that use GMMs for language classification. It also considers how other techniques have been used to improve the performance of systems that use GMMs as basis for language classification.

4.1 Maximum Likelihood Classification Approach

A study conducted by Zissman ranks this type of GMM LID system as the simplest for studying language identification systems [18]. The system structure is illustrated in figure 4.1.

In the training phase, a Gaussian mixture model for the spectral or cepstral feature vectors is created for each language. In the recognition phase, the likelihood of the test utterance feature vectors is computed given each of the training models. The language of the model having the maximum likelihood is hypothesized as the language of utterance. This type of a system is said to perform a static classification, based on the fact that it does not consider the ability to model sequential characteristics of speech [19]. Successive acoustic feature vectors \mathbf{x}_t are assumed to be drawn randomly according to a Gaussian Mixture distribution (GMM), given by

$$p(\mathbf{x}_t|\lambda) = \sum_{j=1}^M \alpha_j \cdot P_j(\mathbf{x}_t)$$

where λ represents the model parameters

$$\lambda = \{\alpha_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}$$

Here the α_j 's are the mixture weights and the P_j 's are the multi-variate Gaussian densities defined by the means $\boldsymbol{\mu}_j$ and the variance $\boldsymbol{\Sigma}_j$. Each language is modelled by a separate GMM. The parameters of each language specific GMM are determined during a training process using the EM algorithm.

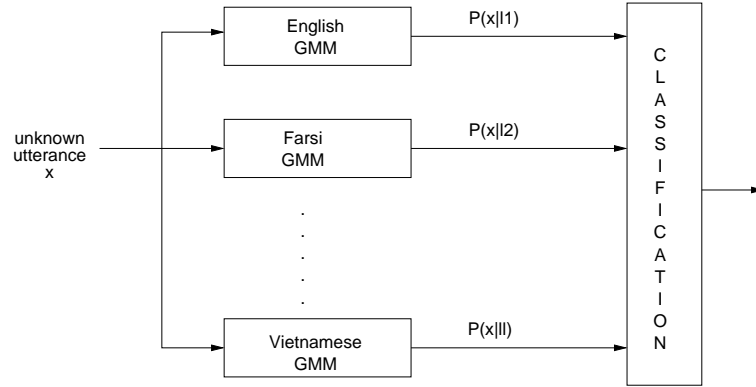


Figure 4.1: LID system based on maximum likelihood classification using GMMs.

4.1.1 Parameterization

In Zissman's implementation of this system two GMMs are created for each language, one for the cepstral feature vectors, $\{\mathbf{c}\}$ and one for the delta-cepstral feature vectors, $\{\Delta\mathbf{c}\}$. From training speech spoken in language l , two independent feature vector streams are extracted every 10ms: Mel-scale cepstra (c_1 through c_{12}) and delta cepstra (Δc_0 through Δc_{12}). Voice Activity Detection based on a time-varying estimate of instantaneous signal-to-noise (SNR) ratio was applied to the speech segments in order to eliminate long periods of silences.

Due to the fact that cepstral features can be influenced by channel effects RASTA¹ was applied to remove slow varying, linear channel effects from the raw feature vectors. The normalised features c' were obtained from the unnormalised features c by convolving with the RASTA filter impulse response

$$c'_i(t) = h(t) * c_i(t)$$

where "*" denotes the convolution. A standard RASTA IIR filter was used

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

4.1.2 Training

A clustering algorithm is applied to cluster each stream of feature vectors, producing 40 cluster centres for each of the two streams.

By using the cluster centres as initial estimates for the means of the GMMs μ_j , multiple iterations of the expectation-maximisation (EM) algorithm are run for each language until an optimised set of α_j , μ_j and Σ_j are obtained.

¹RASTA (relative spectral technique) suppresses the spectral components that change more slowly or quickly than the typical range of change of speech.

4.1.3 Experimental Results

The unknown speech utterance is first parameterised as feature vectors, and subsequently the log likelihood of these features is calculated for each language l . The log likelihood \mathcal{L} is given by

$$\mathcal{L}(\{\mathbf{c}_t, \mathbf{\Delta c}_t\} | \lambda_l^C, \lambda_l^{DC}) = \sum_{t=1}^T [\log p(\mathbf{c}_t | \lambda_l^C) + \log p(\mathbf{\Delta c}_t | \lambda_l^{DC})]$$

where λ_l^C and λ_l^{DC} are the cepstral and delta-cepstral GMM for language l , respectively, while T is the duration of the utterance. Implicit in this equation are the assumptions that the observations $\{\mathbf{c}_t\}$ and $\{\mathbf{\Delta c}_t\}$ are statistically independent of each other, and the two streams are jointly statistically independent of each other.

This approach was tested by performing a two-alternative and a three-alternative classification experiment using English, Japanese and Spanish data from the OGLTS corpus. For the two-alternative set of experiments the "Initial Training" data was used for training and the "Development Test" data for testing. For every pair of languages a pair of GMMs was trained. Messages spoken in the selected pair were passed to the system for classification. For the three-alternative set of experiments the "Initial Training", the "Development Test" and the "Extended Training" data sets were used for training while the "Final Test" set was used for testing. Models were trained in all three languages and test messages in all three languages were presented for classification. The training and development sets referred to are illustrated in Table 4.1. The number of messages per language in each of the four segments are listed, and sub-divided into recordings of male and female speakers. The results of these experiments are shown in Table 4.2. An equal-weighting was given to each language pair.

Language	Initial Training		Development Test		Extended Training		Final Test	
	male	female	male	female	male	female	male	female
English	33	17	14	6	72	30	16	4
Farsi	39	10	15	4	8	1	18	2
French	40	10	15	5	11	2	12	8
German	25	25	11	9	10	5	15	5
Korean	32	17	18	2	3	2	15	5
Japanese	30	20	15	5	1	0	11	8
Mandarin	34	15	14	6	8	8	10	10
Spanish	34	16	16	4	14	5	11	8
Tamil	43	7	17	3	20	2	19	1
Vietnamese	31	19	16	4	11	6	13	7

Table 4.1: Division of the OGLTS Corpus into Training and Test sets.

Some additional experiments were performed using all ten languages of the OGLTS corpus. These results are illustrated in Table 4.3. The first two columns

System	Eng./ Jap.		Eng./Spa.		Jap./Spa.		2L Average		3L Average	
	45s	10s	45s	10s	45s	10s	45s	10s	45s	10s
GMM	17	16	17	16	35	36	23	23	35	36

Table 4.2: Experimental results for various language pairs (% Error).

show the classification for ten languages. The second two columns show the average classification accuracy when distinguishing between English and each of the other nine languages in turn. The last column shows average classification accuracy between the 45 possible language pairs.

System	10L		English vs. L		L vs. L'	
	45s	10s	45s	10s	45s	10s
GMM	47	50	19	16	20	21

Table 4.3: Experimental results when using 10 languages (% Error).

The error rates for 2 language tests are the lowest, but these are the easiest tests since one would expect a 50% error rate if guessing blindly. However, for the 10 language test one would expect a 90% error rate if guessing, which is why the performance is worse (50% vs 23%). The overall performance is somewhat better for longer utterances (45s) than for shorter ones (10s).

4.2 GMM Tokenization Approach

The GMM tokenization system consists of a set of parallel GMMs, each of which is followed by a bank of tokenizer dependent language models. Each tokenizer produces a stream of symbols corresponding to the highest scoring GMM component. The likelihood of each tokenizer dependent symbol stream is evaluated by a set of statistical language models, and the language model scores are fed to the Gaussian back-end classifier for final processing.

The function of the GMM tokenizer is to assign feature vectors to an area of the acoustic space which corresponds to the closest Gaussian component in the mixture model. This is illustrated in figure 4.2. The resulting sequence of tokens are scored by a set of language-dependent language models as shown in the following figure 4.3.

The languages are modelled by bigram models in which the probability of sequences of two consecutive tokens is modelled by the following relation:

$$p(\mathbf{v}_n|\mathbf{v}_{n-1}) = \alpha_2 \cdot P(\mathbf{v}_n|\mathbf{v}_{n-1}) + \alpha_1 \cdot P(\mathbf{v}_n) + \alpha_0$$

where $\alpha_2 = 0.666$, $\alpha_1 = 0.333$ and $\alpha_0 = 0.001$ are fixed constants and \mathbf{v}_n and \mathbf{v}_{n-1} are any two consecutive tokens.

A backend classifier is a GMM that discriminates between the language model scores. In the case of a single GMM tokenizer with N languages, the

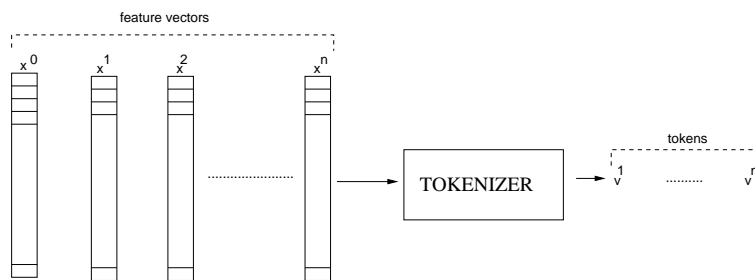


Figure 4.2: Feature vector processing in a GMM tokenizer.

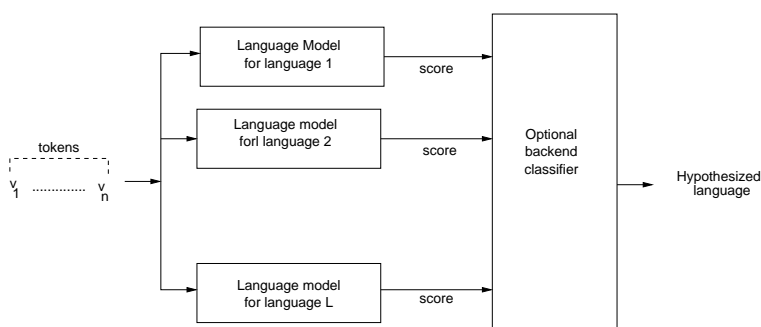


Figure 4.3: A GMM Tokenization system followed by language dependent models.

backend classifier receives an N dimensional vector of language model scores. These can be normalised using linear discriminant analysis, in order to reduce the dimension of the input vector. In the case of multiple tokenizers, this normalisation process decorrelates the information obtained from the different tokenizers. However, a GMM tokenization system can function without a backend classifier, by simply identifying the language of the model with the highest score as hypothesized to be the language of utterance.

In a study carried out by Torres Carrasquillo et al, experiments are carried out to illustrate the varying effects of GMM mixture orders, the use of Backend classifiers, and the combination of language model scores [13].

4.2.1 Parameterization

Evaluations of the GMM tokenization approach to language identification have typically used feature vectors consisting of cepstra and delta cepstra. For conventional cepstra and delta-cepstra $2N$ notation is used, where N is the number of cepstral coefficients computed at each frame. There are 10 cepstral coefficients, and 10 delta cepstral coefficients.

4.2.2 Training

The GMM Tokenizer is typically trained on one language, but is used to decode information for any language it is presented with.

During training, MFCCs are computed every 10 ms (100 per second). The first ten cepstral parameters and their first differential are used. The cepstral vectors are subjected to a RASTA normalisation to remove linear channel effects. Next these feature vectors are used to train a GMM.

4.2.3 Experimental Results

The CallFriend corpus was used in the experimental evaluation. A total of 20 telephone conversations lasting 30 minutes each were used to train the GMM tokenizer and language models, while 1184 utterances lasting 30 seconds each were used to train the Backend classifier. The test set consists of 1492 utterances lasting 30 seconds each.

Single Tokenizer

This GMM LID system consists of a feature extraction pre-processor, a single GMM Tokenizer, a language model for each of the 12 languages and a back-end classifier. This systems structure is illustrated in figure 4.4.

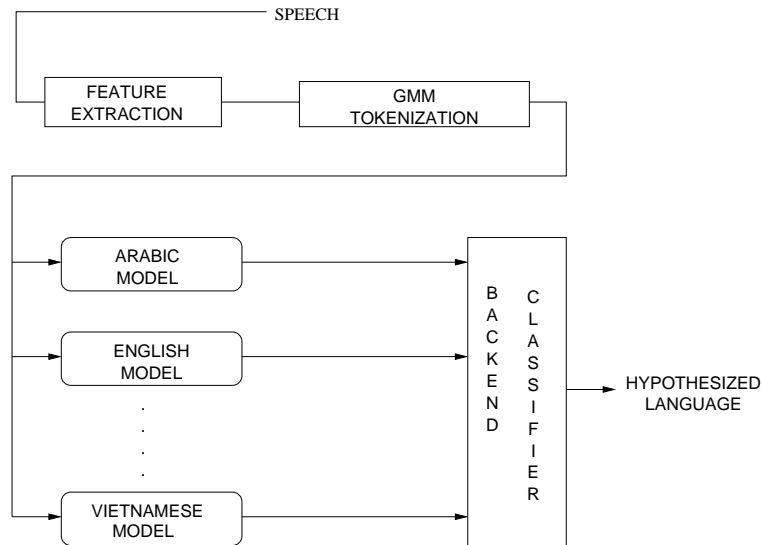


Figure 4.4: Single GMM tokenizer configuration used in [13].

Experiments conducted with the single tokenizer algorithm compare the error rate obtained when using mixture orders ranging from 64 to 512 for the tokenizer GMM.

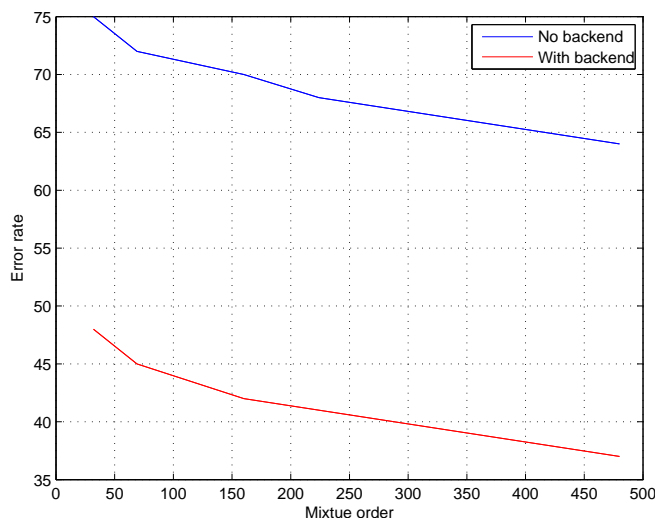


Figure 4.5: Average error rate obtained in [13] when using a single tokenizer for 12-language identification.

A graph showing the average error rate as function of the GMM mixture order for the algorithm that includes a backend Gaussian classifier and another for the error rate without the backend classifier is shown in Figure 4.5. The upper of the two plots shows the performance without the backend classifier. The error rate is reduced by approximately 25% by introducing the backend classifier.

Multiple Tokenizers

Experiments conducted with multiple tokenizers were carried out at a GMM mixture order of 512 and with the inclusion of a backend classifier, given that it produced the lowest error rate when using a single tokenizer.

In order to implement the multiple GMM tokenizer system, each tokenizer is trained with a single language from the corpus.

A graph showing the average error rate as a function of the number of tokenizers is shown in Figure 4.6. The number of tokenizers ranges from one for all 12 languages, to one for each of the 12 languages. This graph shows three plots, the best case scenario, worst case scenario and the average of the two, which are obtained by varying the combination of language tokenizers. The most significant result of this experiment shows that a combination of 4 tokenizers yields the lowest error rate.

The authors of [13] conclude that the performance of a GMM tokenization system is competitive with phone tokenization systems and has the advantage

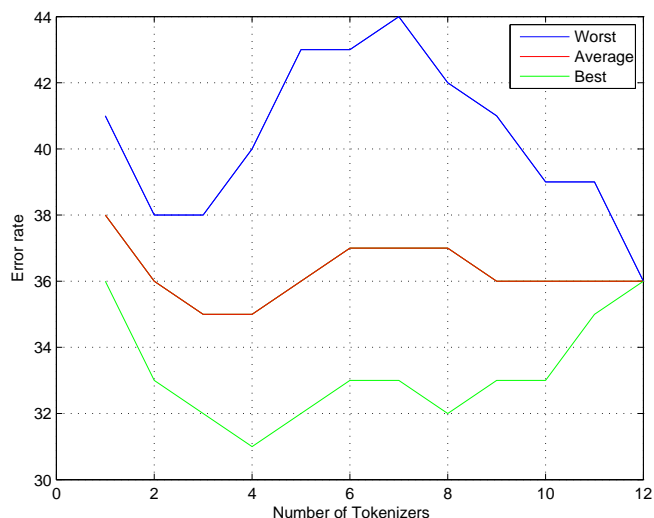


Figure 4.6: Average error rate as a function of multiple tokenizers.

of not requiring transcribed speech. This is also the conclusion of a similar but separate study by the same authors [14].

GMM acoustic scores are generated by the language tokenizers as a by-product of GMM tokenization processing. Consequently, these scores may also be appended to the input vector of the back-end classifier. The results suggest that there is an improvement in LID performance as the GMM order increases, though it is not a very significant improvement.

4.3 UBM Approach

Another type of GMM-based LID system we will consider is described by Wong and Sridharan, which considers the application of a Universal Background Model (UBM). The system structure is illustrated in Figure 4.7.

A Universal Background Model (UBM) is a GMM representing the characteristics of all the different languages to be processed by the LID system. Instead of training language dependent models separately, these are created later by employing Bayesian adaptation from the UBM using the language-specific training speech. Any test observations not seen by the models would typically not discriminate on the bias of any particular LID models.

Previous experiments with GMMs have shown that usually only few of the mixtures of a GMM contribute significantly to the likelihood score for a speech feature vector. In addition, the mixture components of an adapted model of each language share a certain correspondence with the UBM, because each model is

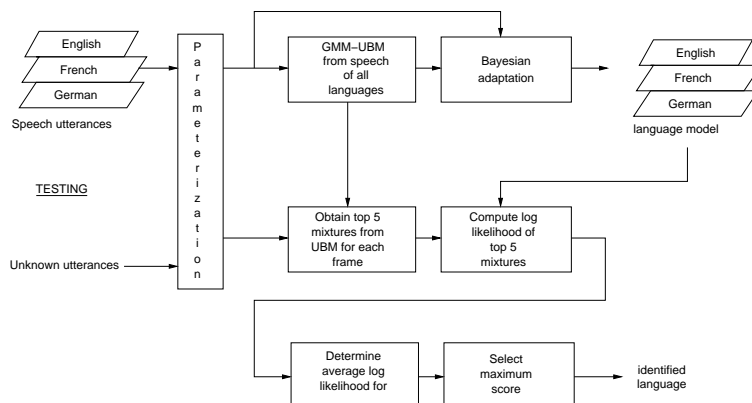


Figure 4.7: LID system based on UBM Adaptation.

adapted from the same UBM. Therefore, the average log-likelihood score for the language-adapted models can be calculated by scoring only the more significant mixtures (for example, the top 5 mixtures). According to the correspondence of mixtures between UBM and the model of a language, these significant mixtures can be obtained by selecting the mixtures from the UBM that have the highest score. By employing this mixture testing strategy, the computation can be reduced significantly.

The UBM technique enables the number of mixtures of the GMM to be increased significantly, as well as the dimension of the feature vector, thereby making it possible to model the characteristics of each language more accurately [16].

4.3.1 Parameterization

In the work by Wong and Sridharan, the feature vectors consist of 12 cepstral coefficients and 12 delta coefficients [15], as described in section 2.1.5.

4.3.2 Training

In order to train the UBM, the trained data from all languages is pooled. Since this increases the training set size, the UBM can be trained to have a higher number of Gaussian Mixtures than GMMs trained on individual languages. Language-specific GMMs are obtained from the UBM by subsequent adaptation or re-estimation.

4.3.3 Experimental Results

The system was trained and tested on the 10 language version of the OGLTS corpus. The results obtained from experiments using UBM indicate that the time required to train and test the LID system is significantly reduced. Due to

this reduction, the number of GMM mixtures can be increased, thereby allowing a more characteristic modeling of the languages.

In Wong's implementation of the GMM-based LID system, the performance of the standard GMM is 56 %, and this compares favourably with Zissman's system for which an accuracy of 50 % was reported. The GMM-UBM system on the other hand has an accuracy of 53 % as is illustrated in Table 4.4.

System	%correct
Standard GMM (Zissman)	50
Standard GMM	56.6
GMM-UBM	53.2

Table 4.4: Comparison of the performance of Standard GMM (Zissman) versus UBM system.

Experiments were also conducted by varying the number of UBM mixtures that we selected in the likelihood calculation. The results obtained indicate that the accuracy decreases slightly as the number of mixtures increase from 1 to 100, and thereafter stays fairly constant. However, this is not a very significant decrease in accuracy. This variation is illustrated in Figure 4.8.

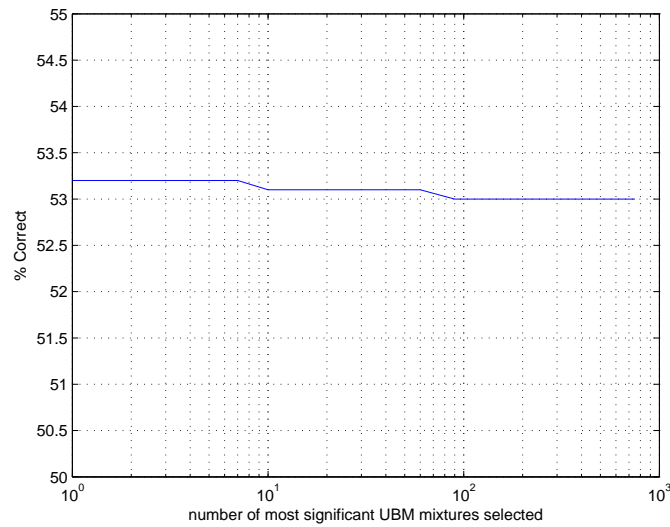


Figure 4.8: Variation of the UBM LID performance with respect to the number of mixtures selected during likelihood computation.

4.4 SDC Approach

The use of Shifted Delta Cepstra (SDC) in a GMM LID system can be viewed as a means of incorporating additional temporal information about the speech into the feature vector. The decision to use temporal information spanning a large number of frames is motivated by the success of phonatic approaches that base their tokenization over multiple frames [14].

4.4.1 Parameterization

The feature vectors were compiled according to the description given in section 2.1.6.

4.4.2 Training

The GMM LID system used in the training and testing process consisted of a frontend feature extraction preprocessor, a GMM for each target language and a backend classifier [14].

In the training process the SDC feature vectors were incorporated into the simple GMM system by replacing the conventional cepstra and delta-cepstra feature vectors. The system used 10-1-3-3 parameterisation with diagonal covariances, and was trained on the CALLFRIEND corpus described in Section 3.3. The training set of the corpus was used to train language models while the development test set was used to train the backend classifier. The structure of the system is the same as the one used in the tokenization approach illustrated in Figure 4.4.

4.4.3 Experimental Results.

Experiments were conducted to compare the performance when using conventional cepstra and when using SDC. 12 languages were used in the experiments, and the evaluation test set of the CALLFRIEND corpus was used to test the performance. The results obtained from the experiments indicate that the performance of the system improves with high-order (≥ 512) GMMs. This is illustrated in Figure 4.9. Note that this figure shows the equal error rate (EER) and can therefore not be compared directly with the error rates presented in Sections 4.1, 4.2 and 4.3.

Equal Error Rate (EER) indicates the point at which the proportion of false acceptances is equal to the proportion of false rejections. The lower the EER the higher the accuracy of the system.

In order to compute the EER, a set of 12 experiments were performed, in each of which the objective was to differentiate between one language and the remaining 11. In each case the number of false positives (when one of the remaining 11 was mistaken for the 12th) and the number of false negatives (when the 12th language was mistaken for the remaining 11) were balanced. An average over all 12 combinations was then calculated and is shown in Figure 4.9.

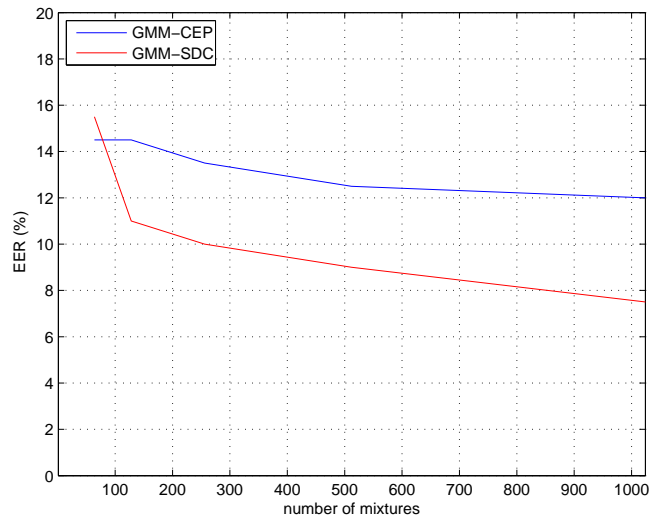


Figure 4.9: Comparison of the performance of a GMM LID system using conventional cepstral features and another using SDC features with respect to number of mixtures.

4.5 Conclusion

GMM LID systems are among the simplest approaches and have the important advantage that they do not require annotated data. They perform static classification, due to the nature of their single state structure, therefore no state transitory probabilities have to be computed.

There are a number of algorithms that can be experimented with to improve the performance of GMM LID systems. A GMM Tokenizer will produce a vector whose features represent the acoustic models of languages. Another algorithm is the UBM that entails cloning language models from a seed model. Finally, SDC features have been shown to outperform the more usual MFCC features by some authors.

Chapter 5

System Development and Evaluation

This chapter will discuss the steps taken to develop language identification systems and to evaluate their performance. Section 1 will describe the generic process that was used in the experiments. Section 2 will focus on systems trained using diagonal covariance GMMs, whilst section 3 focuses on systems trained using full covariance GMMs. Section 4 focuses on systems that use Shifted Delta Cepstra parameterisation, and section 5 focuses on the UBM approach.

5.1 The generic system development and evaluation process

All systems are initialised using hand-picked data selected from the training set. The hand-picked data is assumed to be a good representation of the speech found in the corpus, and it is used to create a seed model with which to initialise the GMM used to model each language. Hand-picked data is also used to initialise a silence model. This is done so that the initial system can discriminate between speech and silence regions in the training data, before it is trained to discriminate between the languages themselves.

After the seed speech and silence model have been created, the language specific models are initialised using the speech seed model. The newly initialised models for the individual languages together with the initialised model for the silence are combined to form a single set of HMMs. At this stage all the language models are identical because they are initialised using the same seed model.

Thereafter, the models are reestimated using the full training set and the different models acquire specialised parameters. These distinct models will make it possible to discriminate between the languages.

Further reestimation of the model parameters allows for a more accurate discrimination between the languages. This is done by iterating the reestimation

process described above, and coupling this with increasing the number of model mixtures.

The mixtures are increased gradually for the different systems. In the case of the diagonal covariance system they are increased in multiples of 2 until 16 mixtures are reached. From 16 onwards they are increased in steps of 16. In the case of the full covariance system the number of mixtures is increased in single steps. After every increase in the number of mixtures the models are reestimated and tested to see how accurately the system identifies the languages. This process is continued in order to identify a point where the improvement in accuracy becomes negligible. The process described above is illustrated in Figure 5.1, and indicates the HTK tools used.

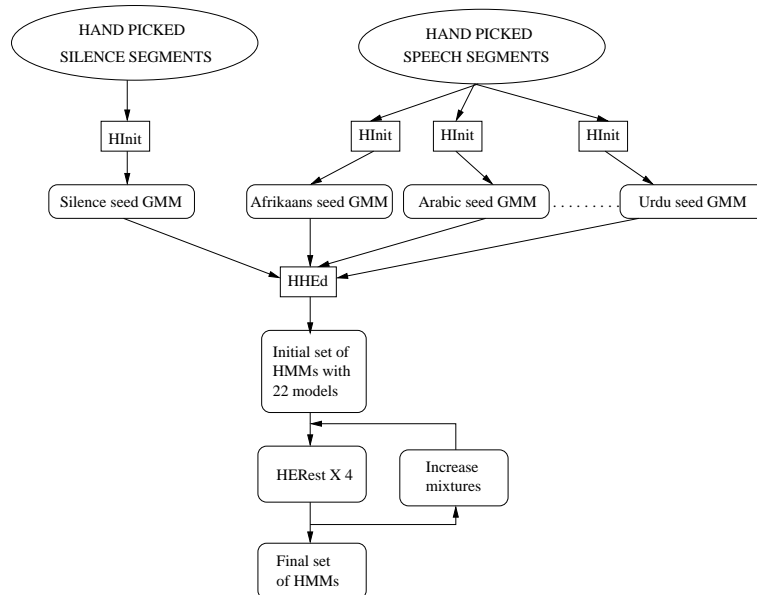


Figure 5.1: Block diagram of the system development process.

In order to test whether the system identifies the languages correctly the transcription of the test set produced by the trained models is compared with the true transcription. The accuracy of identifying the language correctly is computed as a percentage. This process is illustrated graphically in Figure 5.3.

The grammar used in the recogniser is a simple "OR" operator that selects one of 21 languages, as illustrated in Figure 5.3. Each input utterance is terminated by a begin and end silence, and no provision was made for silences within an utterance as care was taken to ensure that utterances were voiced.

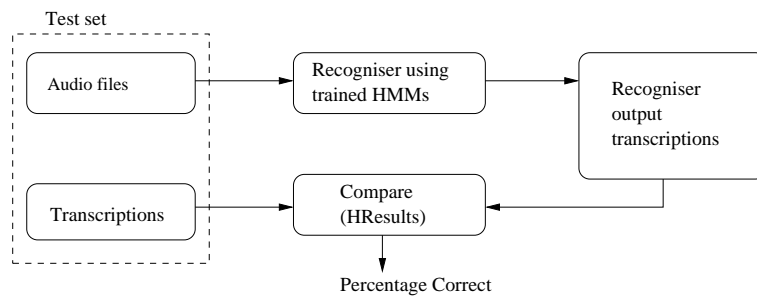


Figure 5.2: Block diagram of the testing process.

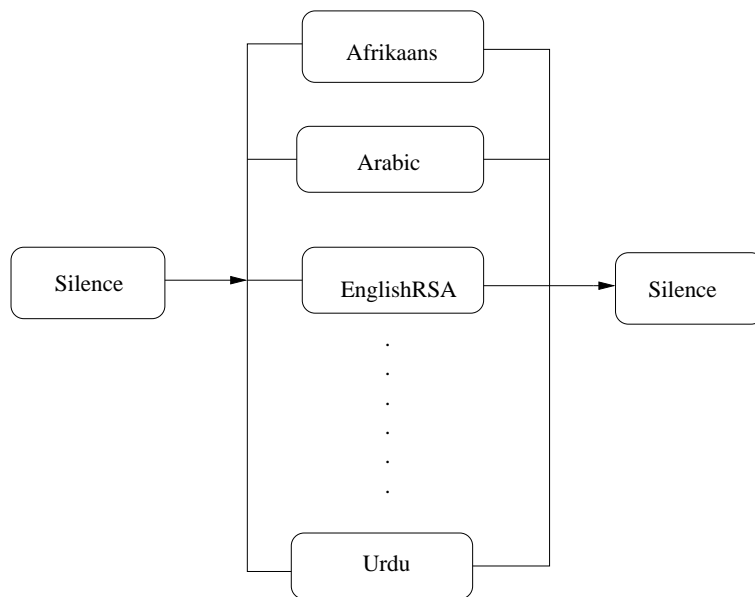


Figure 5.3: Block diagram of the grammar used in the recognition process.

5.2 Diagonal Covariance GMMs

A diagonal covariance GMM assumes all off-diagonal entries of the covariance matrix to be zero. Hence it assumes the individual features in the feature vector to be uncorrelated with each other. Although this assumption does not hold in general, it has the advantage of reducing the number of parameters that must be trained. Both the mean and the covariance of each Gaussian mixture are represented by a $1 \times n$ dimensional vector.

Different parameterisations of the raw speech data can be experimented with in an effort to improve the accuracy of LID systems. The parameterisation type that has been used is MFCC with its first and second differential used

independently and in combination, as described in the following.

5.2.1 MFCC_E

This parameterisation consists of a 1×12 MFCC vector with the energy of each frame appended as a 13th feature.

5.2.2 MFCC_E_D

This parameterisation appends the first differential to the MFCC_E vector resulting in a 26-dimensional vector. The first differential enhances the performance of the system, because it provides transitory information that can be used in discriminating between languages.

5.2.3 MFCC_E_D_A

This parameterisation appends both the first and second differential to the MFCC_E vector, resulting in a 39-dimensional feature vector.

5.2.4 Experimental results

The experiments were performed with the three parameterisations discussed previously. In addition, results were computed for different numbers of mixtures in the GMM. The results are shown in Table 5.1, and graphically in Figure 5.4. The figure shows that the accuracy of identifying the language correctly for a single mixture lies between 9.84 and 10.99%. As the number of mixtures is increased, the accuracy improves. However from approximately 100 mixtures onwards the improvement is small. When the number of mixtures reaches 512 the accuracy lies between 23.16 and 28.46%.

From Figure 5.4 we see that the best performing parameterisation is MFCC_E_D. In particular the results show that the addition of the second differential does not lead to improved performance, but in fact to a deterioration in performance.

No. of mixes	Parameter kinds		
	MFCC_E	MFCC_E_D	MFCC_E_D_A
1	9.84	11.69	10.99
2	9.39	9.67	8.32
4	12.56	11.24	12.43
8	12.77	12.73	14.43
16	14.27	15.32	14.14
32	17.51	19.60	15.82
48	18.55	21.20	17.79
64	19.31	22.75	18.75
80	19.73	22.75	19.29
96	19.90	22.88	2.14
112	19.86	23.79	20.83
128	20.25	23.98	21.72
144	20.31	24.79	21.86
160	20.57	25.51	21.72
176	20.75	25.90	22.31
192	21.01	26.07	22.66
208	21.40	25.96	22.75
224	21.40	26.63	22.85
240	21.29	26.98	22.94
256	21.38	26.94	23.38
272	21.57	27.31	23.64
288	21.81	27.70	23.57
304	21.94	27.89	23.68
320	22.14	28.07	24.05
336	22.27	28.16	24.11
352	22.12	28.03	24.27
368	22.20	28.29	24.33
384	22.33	28.35	24.70
400	22.40	28.00	24.68
416	22.57	28.07	24.66
432	22.70	28.11	24.57
448	22.70	28.20	24.64
464	22.94	28.26	24.77
480	23.27	28.35	24.79
496	23.09	28.59	24.92
512	23.16	28.46	24.85

Table 5.1: The accuracy in percentage of identifying the language correctly for a diagonal covariance system using the MFCC_E, MFCC_E_D and MFCC_E_D_A parameterisation.

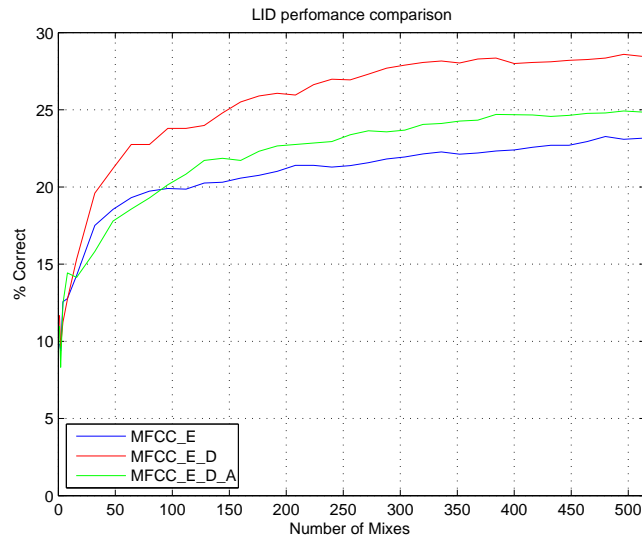


Figure 5.4: Performance of diagonal covariance LID system using MFCC_E, MFCC_E_D and MFCC_E_D_A parameterisation.

5.3 Full Covariance GMMs

A full covariance GMM uses a $n \times n$ upper right triangular matrix to represent the covariance of each Gaussian mixture. This section experiments with the use of these GMMs in all models except the silence model. The silence model in this instance continues to use a diagonal covariance vector. Given that the silence files have little training data, the full covariance model could not be used as this causes numeric problems when reestimating the model parameters.

The experiments were performed using the MFCC_E and MFCC_E_D parameterisations only. Mixtures were increased in steps of 1 due to the much larger number of parameters used by full covariance models. The results obtained are shown in Table 5.2 and illustrated graphically in Figure 5.5. The graph indicates that for a full covariance LID system the accuracy of identifying a language correctly lies between 13.64 and 16.38% with a single mixture per GMM. After increasing the number of mixtures to 64, the accuracy rises to between 24.90 and 28.81%. The results are listed in Table 5.2 and are illustrated graphically in Figure 5.5.

No. of mixes	Parameter kind	
	MFCC_E	MFCC_E_D
1	16.38	13.64
2	17.47	12.10
4	18.92	20.38
8	18.55	22.03
16	21.81	25.57
32	24.20	26.18
48	24.29	28.18
64	24.90	28.81

Table 5.2: The accuracy in percentage of identifying the language correctly for a full covariance system using the MFCC_E and MFCC_E_D parameterisation.

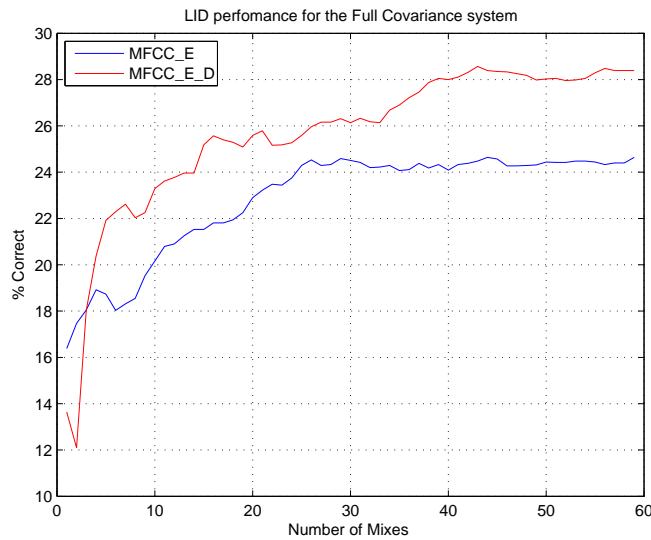


Figure 5.5: Performance of full covariance LID system using MFCC_E and MFCC_E_D parameterisation.

5.4 Shifted Delta Cepstra

The first set of experiments using SDC parameterisation were performed with a 1×30 dimensional vector for both the seed and silence models. 10 MFCC features were used, which means N was set to 10 as defined in Section 2.1.6. The time advance time delay d was set to 1, while the time shift P and the number of concatenated blocks k were equally set to 3. These parameters were found to be optimal in [14].

After having extracted the SDC features, the steps outlined in the generic

system development and evaluation process were applied in training and testing this system. Mixtures were increased in the same increments used in Section 5.1.

The second set of experiments were performed using 12 MFCC features plus an additional energy feature i.e, $N = 13$. The parameters (d , P and k) used to obtain the previous SDC vector remained the same. This results in a 39-dimensional feature vector which can be compared to the 39-dimensional MFCC_ED_A parameterisation used in Section 5.2.3.

The results obtained from these experiments are listed in Table 5.3, and illustrated graphically in Figure 5.6. The graph indicates that there is an improvement in the accuracy of the system as the number of mixtures are increased. The accuracy obtained for a single mixture is 8.28 %. However, from the 16th mixture to the 48th it is evident that this improvement tends to be more linear.

The best performing systems have an accuracy of between 20.07 and 24.20 %. It is also clear that using $N = 13$ instead of $N = 10$ lead to a consistent deterioration in performance.

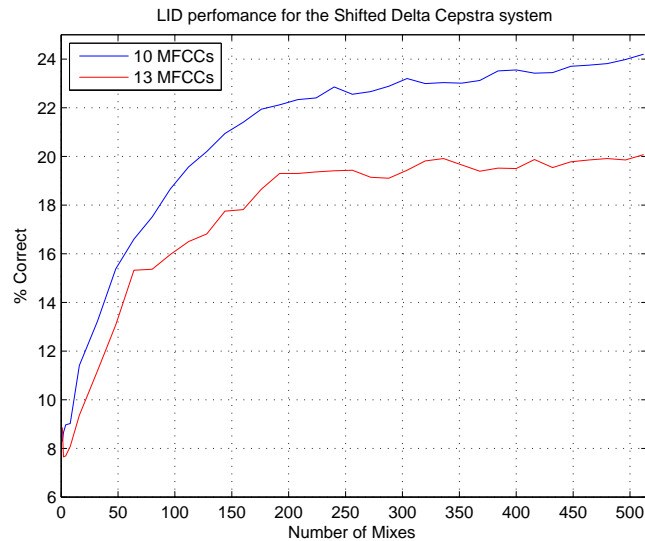


Figure 5.6: Performance of SDC systems based on 10 MFCCs and 13 MFCCs.

5.5 Universal Background Model

The UBM system is again initialised using hand-picked speech and silence data from the training set. The speech data is used to make seed speech model, and the silence data is used to make seed silence model. The two models are then combined to form a single HMM set.

No. of mixes	Parameter kind	
	10 MFCCs	13 MFCCs
1	8.28	8.86
2	8.65	7.65
4	9.02	7.69
8	13.10	8.08
16	11.41	9.36
32	13.23	11.19
48	15.38	13.08
64	16.60	15.32
80	17.51	15.36
96	18.66	15.97
112	19.57	16.50
128	20.20	16.81
144	20.94	17.75
160	21.40	17.81
176	21.94	18.65
192	22.12	19.30
208	22.33	19.30
224	22.40	19.36
240	22.85	19.41
256	22.55	19.43
272	22.66	19.14
288	22.88	19.10
304	23.20	19.43
320	22.99	19.81
336	23.03	19.91
352	23.01	19.65
368	23.12	19.39
384	23.51	19.52
400	23.55	19.50
416	23.42	19.87
432	23.44	19.54
448	23.70	19.78
464	23.75	19.85
480	23.81	19.91
496	23.98	19.85
512	24.20	20.07

Table 5.3: The accuracy in percentage of identifying the language correctly in Shifted Delta Cepstra systems.

Thereafter the model parameters are reestimated and the mixtures are increased in multiples of 2 until 16 mixtures are reached, after which the number of mixtures is increased in increments of 16. Diagonal covariances were used.

For each number of mixtures from 16 onwards, the speech model is cloned for each language, and then adapted to that language by re-estimation on the language-specific part of the training set. This procedure is illustrated in Figure 5.7. Testing proceeds as described in Section 5.1.

Experiments were carried out for MFCC_E, MFCC_E_D and MFCC_E_D_A parameterisations, and for GMM mixture orders up to 512. These results are presented in Table 5.4. Due to time constraints, not all model orders could be trained for each parameterisation.

The table shows that in contrast to the experiments in section 5.2, the addition of acceleration features in the case of the UBM approach leads to a slight improvement in performance. In fact, the 304-mixture MFCC_E_D_A system significantly outperforms the 304-mixture GMM system in Table 5.1.

5.6 GMM to HMM conversion

As a final set of experiments, an attempt was made to convert the best performing GMM systems to HMM models, with the hope of exploiting the transitory information that is contained in these state machines. The reason for this is HMMs possess unique transitory probabilities which can be very useful in discriminating one sequence of feature vectors from another on the basis of sequential dependencies.

It is possible to represent a GMM with an *ergodic*¹ HMM. Each component probability distribution will map to a unique state of the ergodic HMM. Each state will therefore have a single Gaussian component as observation probability distribution. The source GMM mixture weights are used to initialise the transition probabilities as shown in Figure 5.9.

At the top of the figure an HMM with a single emitting state and associated 4 mixture GMM is shown. At the bottom a 4-state ergodic HMM is depicted, with each state corresponding to one of the GMM mixtures. The transition probabilities in the ergodic HMM are initialised as indicated in the figure, and subsequently re-trained using Baum-Welch re-estimation. By subsequent re-training of these transition probabilities, it is hoped that the HMM can better model the temporal correlations of the feature vectors. The GMM assumes feature vectors to occur independently, while HMM introduces temporal transitory relationships. Hence, subsequent parameter estimation was focused on fine-tuning the HMM transition weights.

The GMMs used for our experiments² were those obtained in Section 5.2 using the MFCC_E_D parameterisation, and whose performance was listed in

¹Ergodic HMMs are fully connected HMMs in which every state of the model can be reached from every other state of the model.

²the better performing UBM system in Table 5.4 were not used, since they were not ready at the time of experimentation.

No. of mixes	Parameter kind		
	MFCC_E	MFCC_E_D	MFCC_E_D_A
16	20.73	16.86	20.92
32	21.46	19.55	22.96
48	22.01	25.20	24.25
64	22.62	23.96	25.16
80	22.90	23.96	26.29
96	23.29	25.98	27.33
112	22.99	26.37	27.63
128	23.18	26.31	27.55
144	23.44	26.61	27.40
160	23.57	26.40	28.00
176	23.44	26.92	28.09
192	23.68	26.48	28.24
208	23.77	26.81	28.13
224	23.48	26.87	28.03
240	23.38	27.13	28.68
256	23.53	27.16	28.94
272	23.46	27.26	29.07
288	23.81	27.40	29.11
304	23.70	27.53	29.39
320		27.42	
336		27.29	
352		27.50	
368		27.68	
384		27.79	
400		27.92	
416		28.16	
432		28.11	
448		27.83	
464		28.18	
480		28.35	
496		28.37	
512		28.46	

Table 5.4: The accuracy in percentage of identifying the language correctly in Universal Background Model systems.

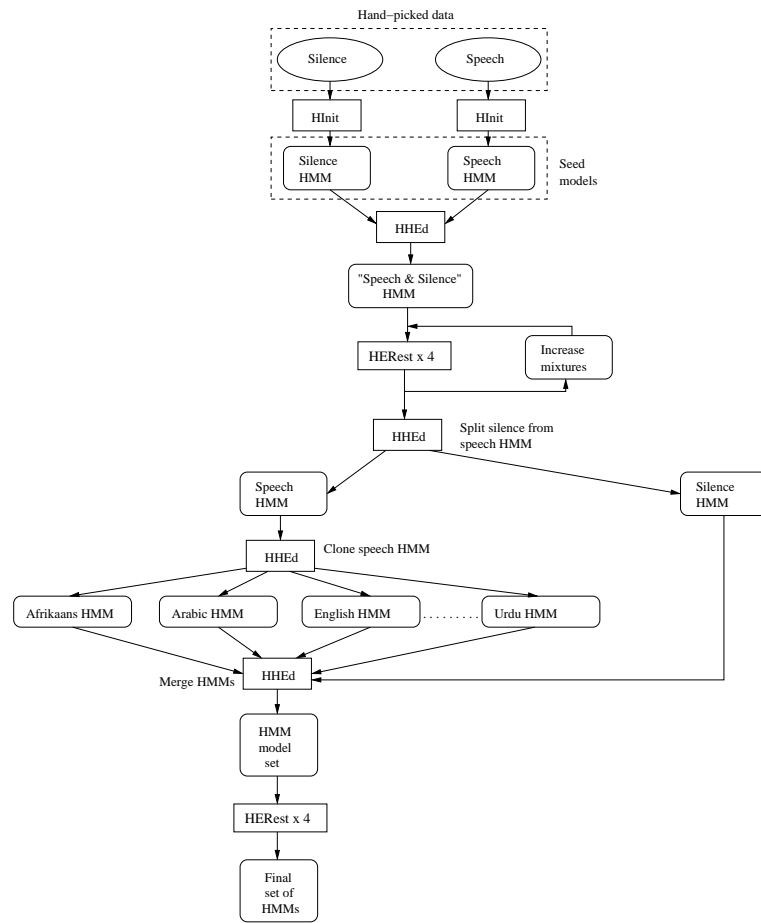


Figure 5.7: Block diagram of the UBM system development and evaluation process.

No. of mixes	GMM system	GMM to HMM system	Average improvement
16	15.32	16.86	10.1
32	19.60	19.55	-0.3
48	21.20	25.20	17.6
64	22.75	23.96	5.3
Average	19.7	29.39	8.2

Table 5.5: Comparison of the accuracy of identifying the language correctly between GMM and GMM to HMM systems.

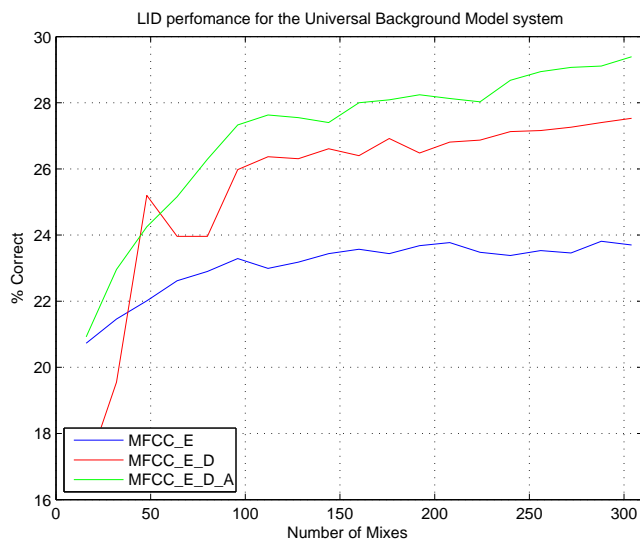


Figure 5.8: The accuracy in percentage of identifying the language correctly in Universal Background Model system.

Table 5.1. Using these as input, Table 5.5 shows the performance of an LID system using HMMs obtained by the GMM to HMM conversion process, followed by 5 iterations of Baum-Welch reestimation to update the transition probabilities.

When comparing results obtained for the GMM system with those of the GMM to HMM system we can see that there is usually a small performance improvement. On average, a relative improvement of 8.2% on the accuracy of the GMM system was achieved. Due to time constraints, experiments for more than 64 mixtures were not possible.

5.7 Error analysis

This final section analyses the errors made by the best-performing LID system; which uses full covariance mixture Gaussians and the MFCC_E_D parameterisation, as was described in Section 5.3. A confusion matrix was determined for this system, which had an overall classification accuracy of 28.81% for 64 mixtures. The confusion matrix is presented in Table 5.6.

The rows of the table correspond to the correct language, while the columns indicate the output of the LID system. For example 19.9% Afrikaans utterances were misclassified as English. Entries on the diagonal indicate the accuracy with which each language was identified individually. The table has been split into two sub-tables for presentation purposes. We see that the best performance is

	Afrikaans	Arabic	Chichewa	English	German	Gujarati	Hindi	Kinyarwanda	Kirundi	Lingala
Afrikaans	25.2	3.2	1.3	1.9	4.4	1.8	1.4	1.7	3.6	0.7
Arabic	5.8	48.6	3.0	8.2	1.5	7.3	23.7	2.8	0.7	1.6
Chichewa	0.0	4.8	46.6	6.9	2.2	1.8	3.6	2.4	1.8	2.3
English	19.9	1.9	0.4	19.5	1.5	7.3	7.2	3.1	1.4	1.0
German	0.7	1.0	2.1	0.0	34.7	0.0	3.6	9.0	1.8	0.0
Gujarati	0.5	5.1	0.4	3.1	0.0	31.2	12.2	0.3	0.4	0.0
Hindi	0.4	2.9	1.3	2.5	7.7	0.0	16.5	2.4	0.7	0.0
Kinyarwanda	0.5	1.0	0.9	0.6	0.7	0.0	0.0	30.9	4.6	1.3
Kirundi	0.5	0.3	0.9	4.4	3.0	0.9	0.7	1.4	37.4	1.0
Lingala	2.1	6.4	1.7	3.8	0.4	7.3	0.7	0.7	2.1	10.7
Luganda	0.2	0.3	4.3	3.8	1.1	0.9	0.7	1.4	0.4	2.0
Nigerian	11.8	4.2	7.3	10.7	0.4	6.4	1.4	2.8	1.1	16.9
Portuguese (Ang)	0.5	2.9	1.7	1.9	1.5	2.8	11.5	1.0	0.0	2.9
Portuguese (Moz)	1.1	1.3	0.9	0.0	0.4	1.8	0.0	1.0	3.2	2.0
Russian	4.6	1.6	2.1	0.0	28.4	0.0	6.5	6.9	3.2	0.0
Shangaan	9.3	1.3	1.7	6.3	1.5	5.5	0.0	1.0	0.7	2.3
Shona	0.4	3.9	15.4	5.7	3.0	0.9	3.6	2.1	1.1	10.7
Sotho	10.1	0.3	4.3	2.5	5.9	0.0	1.4	19.8	27.8	0.0
Swahili (DRC)	4.8	2.6	2.6	6.9	1.1	9.2	1.4	5.6	7.1	40.1
Swahili (Tza)	0.9	0.6	0.4	11.3	0.4	4.6	2.2	2.8	0.7	4.6
Urdu	0.7	5.8	0.9	0.0	0.4	10.1	1.4	0.7	0.4	0.0

Table 5.6: Confusion matrix for best best performing LID system. Columns indicate correct language, while rows indicate the classification made by the LID system.

	Luganda	Nigerian	Portuguese (Ang)	Portuguese (Moz)	Russian	Shangaan	Shona	Sotho	Swahili (DRC)	Swahili (Tza)	Urdu
Afrikaans	4.7	2.3	7.7	4.7	14.4	0.7	0.7	2.0	1.4	4.6	3.6
Arabic	3.1	7.8	4.2	15.7	7.4	0.7	1.8	1.7	5.6	6.6	10.3
Chichewa	15.5	3.9	0.0	4.7	1.1	0.7	7.4	7.5	4.9	2.0	1.2
English	0.0	0.8	2.8	7.0	2.1	2.9	4.3	3.1	2.1	4.6	5.5
German	1.6	3.1	4.2	1.2	3.2	1.4	1.1	6.1	0.0	7.9	0.6
Gujarati	3.9	1.6	1.4	6.4	3.2	1.4	1.4	2.0	0.0	9.3	6.1
Hindi	1.6	0.0	1.4	1.2	0.0	0.7	0.4	0.3	0.7	0.0	7.9
Kinyarwanda	3.1	1.6	4.9	1.2	0.0	0.7	5.3	2.4	2.8	0.7	2.4
Kirundi	0.8	0.0	0.7	1.2	0.0	0.7	2.8	1.7	1.4	0.7	2.4
Lingala	3.1	9.4	4.2	2.9	2.1	0.0	8.5	3.1	13.2	8.6	9.7
Luganda	16.3	1.6	1.4	1.7	0.0	0.7	1.8	3.7	2.1	12.6	0.6
Nigerian	3.9	39.8	2.1	12.8	2.1	12.9	9.9	3.1	2.1	8.6	6.1
Portuguese (Ang)	3.1	0.8	23.8	2.9	2.1	0.0	0.4	2.0	2.1	5.3	0.0
Portuguese (Moz)	0.0	0.8	8.4	10.5	1.1	0.7	5.3	1.0	0.7	2.0	3.6
Russian	4.7	2.3	5.6	1.7	45.7	0.7	1.8	7.8	0.7	2.6	0.6
Shangaan	4.7	10.9	1.4	9.3	0.5	7.9	8.9	4.4	6.9	2.0	9.1
Shona	8.5	2.3	14.0	7.0	3.2	20.0	14.2	2.0	7.6	4.0	2.4
Sotho	4.7	0.0	4.9	0.0	1.6	7.9	8.5	35.9	4.9	4.6	0.0
Swahili (DRC)	13.2	6.2	2.8	2.9	9.6	3.6	11.7	8.1	31.9	2.6	3.6
Swahili (Tza)	3.1	4.7	1.4	3.5	0.0	3.6	3.5	1.0	6.2	7.3	4.8
Urdu	0.8	0.0	2.8	1.7	0.5	2.1	0.4	1.0	2.8	3.3	19.4

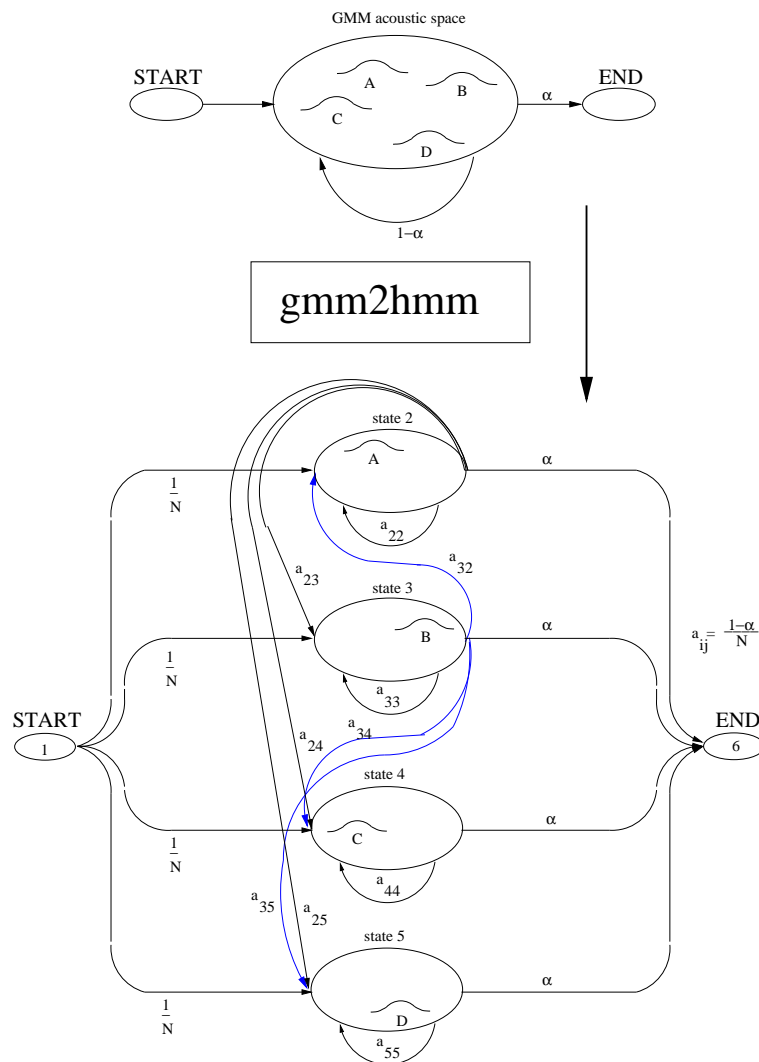


Figure 5.9: Diagrammatic representation of the GMM to HMM model conversion process. The possible paths that the transition from one state to the next are shown with the interconnecting lines labeled a_{ij} indicating the state it is coming from i and the state it is going to j . Only the transitions of state two and three are shown to avoid clutter, however the paths for states four and five adhere to the same principles.

achieved for Arabic and Chichewa, which can be identified with accuracies of 48.6% and 46.6% respectively. The worst performance is achieved for Shangaan and Tanzanian Swahili, with accuracies of 7.9% and 7.3% respectively. Furthermore the following observations can be made [3].

- Congolese Swahili are often confused, which may be due to the fact that they are both spoken in the Democratic Republic of Congo (DRC).
- Urdu and Hindi are both Indo-Aryan languages, and are often confused.
- English, Afrikaans and German are all West-Germanic languages. The first two are often confused.
- Nigerian can be seen as a set of more than 10 sub-languages, and therefore diverse. This may explain why it is confused with a variety of languages.
- Chichewa, Kinyarwandi, Kirundi, Lingala, Luganda, Shangaan, Shona and Swahili all belong to the Southern Bantu group of languages. The table shows that there is a fair degree of confusion among these languages, for example between Lingala, Shona and Swahili.

Language Family	No. of languages	Identification accuracies %
Germanic	3	43.07
Latin	2	35.85
Slavic	1	45.7
Indo-Aryan	3	34.93
Semetic	1	48.6
Southern Bantu	11	75.69

Table 5.7: Identification accuracy within language families.

Finally, the identification accuracy within each language family can be calculated. These accuracies are presented in Table 5.7. The highest for the Southern Bantu languages, which is also represent the largest group in the SSLC corpus, and therefore also the largest portion of training data. Latin and Indo-Aryan languages are the least easy to identify.

5.8 Summary

The identification accuracy achieved by the LID systems developed in this chapter lies between 8 and 30% . The highest scores were achieved by the Diagonal Covariance and Full Covariance systems, scoring a high of 28.46 and 28.81% respectively. In both cases the use of MFCC's, energy and their first differential, but not their second differential, led to optimum performance. However, when employing the UBM approach, the addition of acceleration coefficients does lead to performance gains.

The introduction of transition probabilities by means of the GMM to HMM conversion in general also lead to further improvements for models of the same order. Finally, the uses of shifted delta cepstra did not lead to improvements over the use of MFCC features.

Chapter 6

Summary and conclusions

In this thesis, the aim of developing a LID system for African languages that is based on simple stochastic models has led to the implementation of various approaches.

The first phase of the work described in this thesis dealt with the compilation of a corpus containing languages commonly spoken in Southern Africa. The resulting SSLC corpus contained data from 4772 speakers in 21 languages, and contains a total of approximately 80 hours of speech.

The use of GMM-based models has the advantage that no transcribed data is required. This is an important consideration for languages in the Southern African region, which are generally not technologically developed, and for which transcription are not available.

The use of GMMs in various configuration and using various MFCC-based parameterisations was evaluated. It was found that increasing mixtures led to a general improvement, but levelled out above 300 mixtures. For single GMM systems MFCC_ED gave the best performance. However when systems are trained using a UBM, small further improvements are achieved by also including acceleration coefficients. Using full covariance did not improve on use of diagonal covariance when number of parameters increased. Finally the GMM to HMM conversion strategy led to a slightly better performance, but remains to be tested for large number(512) of mixtures.

The best performing systems were based on either diagonal or full-covariance GMMs using features together with their first differential. The best performance of 29.39% language identification accuracy compares favourably with results reported in the literature. For example, in a 12-language experiment by Torres-Carrasquillo et al [13] an identification accuracy of 35% was achieved for a similar number of mixture components. Although our best performance is slightly lower, it must be borne in mind that our system was dealing with a larger number of languages.

Finally, the introduction of transition probabilities led to a relative performance improvement of approximately 8% on average for diagonal-covariance systems. Unfortunately time did not allow for experimentation with higher

model orders.

Open source tools have been used throughout this work, and not only make the developed system economical as no licence costs are incurred, but also easier to implement because these tools are already compatible with some open source platforms based on the UNIX system like Linux.

Chapter 7

Recommendations and future work

The systems developed in this thesis represent a baseline for further work, since even the best performance achieved is too low for practical use. The following additional steps could lead to improvements:

1. A backend classifier should be introduced, as described in Section 4.2. This means that LID decisions should not just be based on the highest GMM score. Instead, a classifier is trained using these scores as inputs. This has been found in the literature to lead to a further 25% absolute improvement in identification accuracy.
2. The shifted delta cepstra should be augmented with the GMM acoustic likelihoods, and then fed to a backend classifier. In the literature this configuration leads to improvement over the use of MFCCs, which were not achieved in our experiments.
3. For comparative purposes, some phone-recognition based systems, such as PPRLM, should be implemented. This would allow it to be determined whether such approaches do in fact lead to better performance than GMM-based systems for our set of languages
4. The GMM to HMM conversion should be tested for higher model orders (i.e. 512) to establish whether this approach leads to improvements also for our best performing systems.
5. The SSLC corpus could be improved by
 - (a) Removing the macrolanguage "Nigerian" and labelling the individual languages instead. However this requires rare linguistic knowledge.
 - (b) Including more Southern African languages, such as isi Xhosa and isi Zulu.

Bibliography

- [1] Bilmes, J. A., “A Gentle Tutorial of the EM algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models.” *ICSI Technical Report*. TR-97-021.
- [2] ldc. <http://www.ldc.uppen.edu/Catalog/ByType.jsp\#speech>, LDC96S46–LDC96S60. Online source of Linguistic Data Consortium’s information.
- [3] Lewis, M. P., *Ethnologue: Languages of the World*. 16th edition edition. SIL International, 2009.
- [4] Liberman, M. and Cieri, C., “The Creation, Distribution And Use Of Linguistic Data.” in *Proceedings of LREC*, (Granada, Spain), May 1988.
- [5] McLachlan, G. J. and Krishnan, T., *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics, May 2004.
- [6] Muthusamy, Y. K., Barnard, E., and Cole, R. A., “Reviewing Automatic Language Identification.” *IEEE Signal Processing Magazine*, October 1994, Vol. 11, pp. 33–41.
- [7] Muthusamy, Y. K., Cole, R. A., and Oshika, B. T., “The OGI Multi-Language Speech Corpus.” in *Proceedings of ICSLP*, (Banff, Alberta, Canada), October 1992.
- [8] ogi. <http://cslu.cse.ogi.edu/corpora/mlts/protocol.html>. Example prompts used in the OGLTS data acquisition.
- [9] Peebles, P. Z., *Probability, Random Variables And Random Signal Principles*. 4th edition edition. McGraw-Hill International Edition, 2001.
- [10] Rabiner, L., “A Tutorial on hidden Markov models and Selected Applications in Speech Recognition.” in *Proceedings of IEEE*, (Murray Hill, New Jersey, USA), February 1989.
- [11] Rabiner, L. and Juang, B.-H., *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

- [12] Schultz, T., “GlobalPhone, A Multilingual Speech And Text DataBase Developed At Karlsruhe University.” in *Proceedings of ICSLP*, (Denver, Colorado), September 2002.
- [13] Torres-Carrasquillo, P. A., Reynolds, D. A., and Deller, J. R., “Language Identification Using Gaussian Mixture Model Tokenization.” in *Proceedings of ICASSP*, (Orlando), May 2002.
- [14] Torres-Carrasquillo, P. A., Singer, E., Kohler, M. A., Greene, R. J., Reynolds, D. A., and Deller, J., “Approaches to Language Identification Using Gaussian Mixture Models And Shifted Delta Cepstral Features.” in *Proceedings of ICSLP*, (Denver, Colorado, USA), September 2002.
- [15] Wong, E., Pelecanos, J., Myers, S., and Sridharan, S., “Language Identification Using Efficient Gaussian Mixture Model Analysis.” *Speech Research Lab, RCSAVT*.
- [16] Wong, E. and Sridharan, S., “Methods to Improve Gaussian Mixture Model Based Language Identification System.” in *Proceedings of ICSLP*, (Denver, Colorado, USA), September 2002.
- [17] Young, S., Evermann, G., and Hain, T., *The HTK Book*. Cambridge University Engineering Department, December 2002.
- [18] Zissman, M. A., “Comparison of Four Approaches to Automatic Language Identification of Telephone Speech.” *IEEE Transactions on Speech and Audio Processing*, January 1996, Vol. 4, No. 1, pp. 31–44.
- [19] Zissman, M. A. and Berkling, K. M., “Automatic Language Identification.” *Speech Communication*, 2001, No. 35, pp. 115–124.