UNIVERSAL SYLLABLE BASED LANGUAGE IDENTIFICATION SYSTEM

A THESIS

submitted by

SUBHADEEP DEY

for the award of the degree

of

MASTER OF SCIENCE

(by Research)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

January 2012

TABLE OF CONTENTS

List of Tables **v**

List of Figures **vi**

1 Introduction **1**

1.1 Main Contribution of the thesis 3

1.2 Thesis Outline 4

2 Related Work and Background **5**

2.0.1 Probabilistic Approach to Language Identification 5

2.0.2 Architecture of LID systems 7

2.1 Tokeniser 8

2.2 Features for Language Identification 9

2.2.1 Mel Frequency Cepstrum Coefficient (MFCC) 9

2.2.2 Shifted Delta Cepstrum (SDC) 10

2.2.3 Mel Slope 10

2.2.4 Group Delay Features 11

2.3 Class Model 12

2.3.1 Acoustic Model 12

2.3.2 Language Models/Phonotactics 12

2.4 Evaluation Metric 12

2.5 Literature Review 13

2.5.1 1990-2000 14

2.5.2 2000-present 16

2.5.3 Baseline segmentation algorithm 21
2.5.4 Baseline Syllable Clustering Algorithm 22
2.5.4.1 Initial Cluster Selection 23
2.5.4.2 Incremental Clustering 23
2.5.5 Testing 24
3 Unsupervised clustering of syllables for Language Identification 25
3.0.6 Drawback of the baseline clustering approach 25
3.1 Top Down Clustering 26
3.1.1 Computation simplification 27
3.1.2 Algorithm to built syllable models in top down fashion 28
3.2 Universal Syllable Model (USM) 29
3.2.1 Single Universal Syllable Model (Single USM) 30
3.2.2 Multiple Universal Syllable Model (Multiple USM) 30
3.3 Experiments 32
3.3.1 Baseline Syllable based LID system 34
3.3.2 Top Down Clustering 35
3.3.3 Single Universal Syllable Model 36
3.3.4 Multiple Universal Syllable Model 36
3.3.5 Discussion 37
3.4 Conclusion 37
4 Spoken Language Identification using Phonotactics 38
4.1 Phonotactics 38
4.2 Probabilistic framework for incorporating phonotactics information . . 38
4.3 Baseline GMM Tokeniser using SDC features 40
4.4 Language Identification using Language Model 42

4.4.1 Obtaining N-gram statistics . . . 42
4.4.2 Testing Phase . . . 43
4.4.3 Histogram approach . . . 43
4.4.4 Vector Space approach . . . 43
4.5 Multiple Tokeniser Based Approach . . . 44
4.6 Experiments . . . 45
4.6.1 Baseline GMM-UBM system using SDC features . . . 45
4.6.2 Universal Syllable Model . . . 48
4.6.3 Combining System . . . 48
4.7 Conclusion . . . 49
5 IVector for Language Identification 50
5.1 Introduction . . . 50
5.1.1 Parameter Estimation . . . 51
5.1.2 Scoring . . . 52
5.1.3 Linear Discriminant Analysis (LDA)and Within Class Covariance Matrix (WCCN) . . . 53
5.1.4 Integrating syllable based system in ivector framework . . . 53
5.1.5 Experiments . . . 54
5.1.5.1 ivector system . . . 54
5.1.5.2 Syllable based system . . . 54
5.2 Feature Switching Experiments . . . 55
5.2.1 Utilising information from multiple feature streams . . . 56
5.2.2 Exploiting feature diversity . . . 56
5.2.3 Issues in implementing feature-switching . . . 57
5.2.3.1 Application to verification . . . 57

5.2.4 Determining the optimal-feature for a class 57
5.2.5 Verification framework 58
5.2.6 Experiments 58
5.2.6.1 Baseline results 59
5.2.6.2 Implementing feature-switching 59
5.3 Conclusion 61
6 Conclusion **62**
Bibliography **63**

LIST OF TABLES

3.1 *Performance of LID systems on OGI-MLTS and NIST 2003 LRE.* . . . 34

4.1 *Performance of LID systems on OGI-MLTS* 47

5.1 *Performance of LID systems on NIST 2005 LRE* 54

5.2 Verification EERs for the three tasks. Early fusion results of some feature combinations are also given. 60

5.3 Equal error rates for the three tasks using feature-switching. In some cases, feature-switching is done between early fusion of features. 61

LIST OF FIGURES

2.1 Overview of LID system 7

2.2 Different levels of feature extraction in LID task 8

2.3 *Extracting MFCC Features from a frame of speech.* 9

2.4 *Extracting of SDC Features from a frame of speech.* 10

2.5 DET Curve 13

2.6 *Segmentation of an utterance taken from OGI Database. The peaks in the second plot gives the location of the syllable boundaries.* 22

3.1 Obtaining syllable models by top down clustering. 26

3.2 Multiple Universal Syllable Model Approach: Red ellipses represent constant density curves for the GMMs of universal syllable model, green and blue ellipses correspond to GMM of English and Farsi language syllable models respectively obtained after adaptation. 31

3.3 Obtaining Universal Syllable models from training data 33

4.1 *General structure of LID system.* 41

4.2 Multiple Tokeniser used to decode speech utterance. 46

5.1 The proposed verification system incorporating feature switching. (a) Training phase and (b) testing phase. 59

CHAPTER 1

Introduction

Automatic Spoken Language Identification (LID) is the task of identifying an utterance as belonging to one of the languages known previously. The process should be independent of the speaker, accent variation, channel and environment variation. The application of LID systems are two folds: **(a)** as front end for humans and **(b)** as front end for machines. As front end for humans, it can be used to direct the call to an appropriate person who knows the language; as front end for machines it can be used as speech to speech translation. In absence of a LID system, all the speech recognizers have to be run parallely to decode the input utterance which is computationally demanding in a real application. Moreover in a country like India where 22 languages are spoken and thousands of dialectal variation exists, it can be major help. A front end LID system can be used to predict top scoring or a few languages that matches the spoken utterance.

To build automatic LID system, language specific information has to be extracted from the speech signal. Perpetual experiment was done in [1] to identify the main sources of LID features. These distinguishing characteristics are summarized below:

- **Phoneme inventory**: Although the set of phonemes are same among different language, however their acoustic manifestation and duration of these units varies among languages. The frequency of phonemes differs among languages.
- **Prosody**: Languages vary in terms of the duration of phones, speech rate and

the intonation (pitch contour). Tonal languages (i.e. languages in which the intonation of a word determines its meaning) such as Mandarin and Vietnamese have very different intonation characteristics than stress languages such as English. Humans use a variety of cues for language identification, such as sing-song (Mandarin), harsh (German) etc. We expect the prosody to explain/capture such characteristics.

- **Phonotactics** : Phonotactics is the set of rules in a language that provides different combination of phones in a language. For example, the phone cluster /sr/ is very common in the Dravidian language Tamil, whereas it is not a legal cluster in English.
- **Vocabulary**: At gross level, the difference among languages lies in the different words used to express meaning.

These information have been used to built LID systems [2] to a great success. The most successful LID system uses phones extensively to achieve low error rates [3] as reported in the literature. However, the problem with phone based LID system is that it needs labeled corpora to build phoneme models. In this work, the focus is on techniques to language identification which does not need annotated data. The other problem with phone based LID system is that phones by itself cannot distinguish languages as they share the same phoneme inventory. Hence, a longer sub-word unit like syllable contains the language distinguishing characteristics as the number of unique syllables between languages are very high.

Most of the LID systems [4] [3] build statistical models with out explicitly capturing the acoustic characteristics of the units that make up a language. These systems build statistical models that capture the unique characteristics of the language using huge amount of data. The state of the art system [5] uses Call Friend corpus to build the

initial models for language identification. The Call Friend corpus is about 800 hours of data. It is conjectured that humans do not require such amount of data for identifying languages and language identification relies on signal processing cues to identify events [1] and optionally followed by statistical modeling. Since syllable is the basic unit of production, a hybrid model has to used where syllable boundaries are obtained using knowledge based signal processing. The recognition of syllables is performed using isolated-style Hidden Markov Model (HMM). In this thesis, syllable based LID system is used as the baseline and its problem has been addressed.

1.1 Main Contribution of the thesis

The main contribution of the thesis are the following:

- The syllable based LID system has been studied extensively. The performance of the baseline syllable based system [6] has been studied in NIST LID databases, namely OGI-MLTS, NIST 2003 and NIST 2005 LRE. The shortcomings of the technique has been studied and two techniques has been proposed to remove the shortcomings of the technique.
- Finally, the i-vector has been explored in the context of language identification. An attempt has been made to integrate our proposed technique with the i-vector framework.
- The concept of "feature diversity" has been explored in the context of language identification.

[1] Events can be syllable, word boundaries.

1.2 Thesis Outline

The thesis has been organized as follows: chapter 2 describes the related work and the baseline syllable based LID system. In chapter 3, the problem with the baseline system has been highlighted and these problems are addressed by building the syllable models in (a) top down fashion, and (b) first building common syllable models and the language models are then obtained by adaptation. Chapter 4 explores syllable based phonotactics as opposed to Gaussian Mixture Model (GMM) tokenizer. In section 5, the successful speaker verification technique, namely i-vector has been compared to the syllable based LID system. Finally section 6 concludes the report.

CHAPTER 2

Related Work and Background

The language identification problem is usually posed as a pattern recognition problem and consists of two phases (a) Training: Building language/class model, and (b) Testing: scoring against the models and pick the highest scoring language. During the training phase, various distinctive features like prosodic, acoustic etc. are extracted from the speech signal. Statistical models are build using these features. During testing, these information/features have to be combined in a novel way to predict a language. In [7], a probabilistic framework has been proposed to incorporate the various cues viz acoustic, phonotactics, prosodic information, etc.

2.0.1 Probabilistic Approach to Language Identification

The problem of combining multiple information sources is described as:
Given acoustic features $\boldsymbol{X}$, sequence of phone units $\boldsymbol{a} = \{ \boldsymbol{a_1}, \boldsymbol{a_2}, \cdots, \boldsymbol{a_2} \}$ and the prosodic features $\boldsymbol{f} = \{ \boldsymbol{f_1}, \boldsymbol{f_2}, \cdots, \boldsymbol{f_N} \}$ for an utterance, the problem reduces to finding the language (L_i) given these information in maximum likelihood sense. Mathematically it is formulated as:

$$\boldsymbol{L_i'} = \underset{i}{argmax}[p(\boldsymbol{L_i}|\boldsymbol{X})] \tag{2.1}$$

The last equation can be expanded out to incorporate various information like phonotactics, prosodic as follows:

$$\boldsymbol{L_i^{'}} = \underset{i}{argmax} \sum_{a,f} [p(\boldsymbol{L_i}, \boldsymbol{a}, \boldsymbol{f}|\boldsymbol{X})] \tag{2.2}$$

$$\boldsymbol{L_i^{'}} = \underset{i}{argmax} \sum_{a,f} [p(\boldsymbol{L_i}, \boldsymbol{a}, \boldsymbol{f}|\boldsymbol{X})] \tag{2.3}$$

The joint probability $p(\boldsymbol{L_i}, \boldsymbol{a}, \boldsymbol{f}|\boldsymbol{X})$ can be factorised into $p(\boldsymbol{L_i}|\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{X})\ p(\boldsymbol{a}|\boldsymbol{X})\ p(\boldsymbol{f}|\boldsymbol{X})$ and can be written as:

$$\boldsymbol{L_i^{'}} = \underset{i}{argmax} \sum_{a,f} [p(\boldsymbol{L_i}|\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{X}) p(\boldsymbol{a}|\boldsymbol{X}) p(\boldsymbol{f}|\boldsymbol{X})] \tag{2.4}$$

As seen from equation 2.4 the most likely language is to be obtained by optimizing over all the possible segments and phone units which is computational inefficient. If these phones segment are obtained using a speech recognizer. The above equation is significantly simplified as it does not iterate over all possible phone units. The language identification problem reduces to:

$$\boldsymbol{L_i^{'}} = \underset{i}{argmax} [p(\boldsymbol{L_i}|\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{X}) p(\boldsymbol{a}|\boldsymbol{X}) p(\boldsymbol{f}|\boldsymbol{X})] \tag{2.5}$$

Taking logarithm of equation 2.7 the problem reduces to linear combination of the various models as

$$\boldsymbol{L_i^{'}} = \underset{i}{argmax} [log(p(\boldsymbol{L_i}|\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{X})) + log(p(\boldsymbol{a}|\boldsymbol{X})) + log(p(\boldsymbol{f}|\boldsymbol{X}))] \tag{2.6}$$

where

- $p(\boldsymbol{L_i}|\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{X})$ is the prior probability of the language L_i
- $p(\boldsymbol{a}|\boldsymbol{X})$ is the phonotactic model of language L_i.
- $p(\boldsymbol{f}|\boldsymbol{X})$ is the prosodic model of language L_i.[1]

[1]Prosodic features and models has not been studied in this thesis

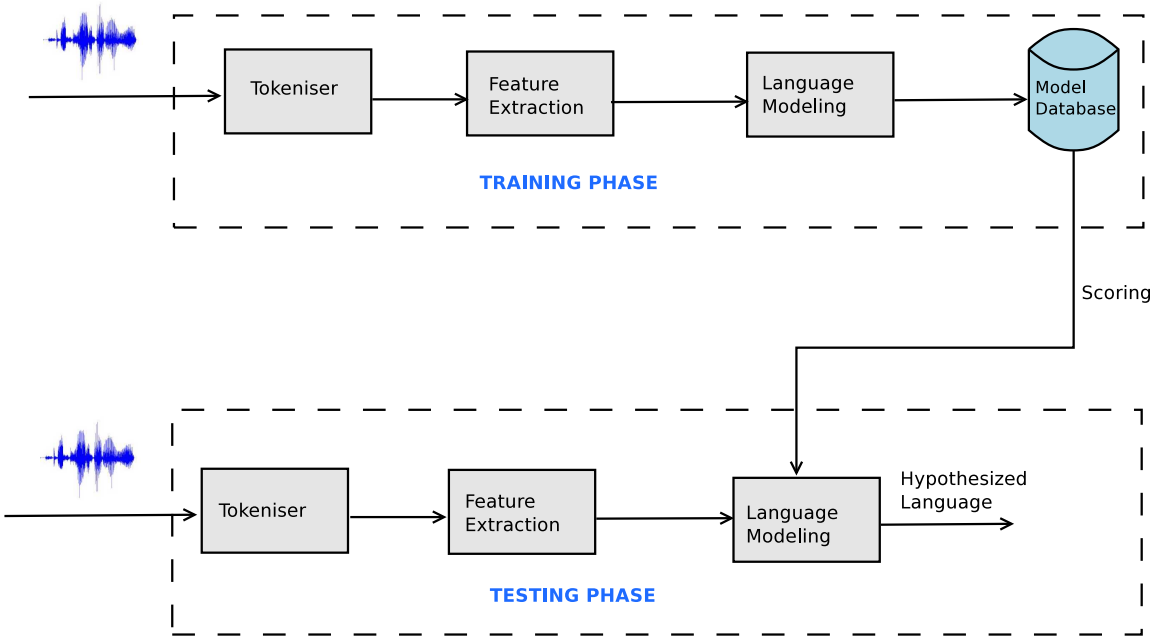

Fig. 2.1: Overview of LID system

The above equation provides a simple framework to integrate various knowledge sources. As can be seen from equation 2.6, the term to be maximized is a linear combination of various information sources with equal weightage. This framework can be generalized by giving appropriate weightage to these cues:

$$L_i^{'} = \underset{i}{argmax}[\alpha \log(p(\boldsymbol{a}|\boldsymbol{X})) + \beta \log(p(\boldsymbol{f}|\boldsymbol{X})) + \gamma \log(p(\boldsymbol{X}|L_i))] \tag{2.7}$$

where $\alpha + \beta + \gamma = 1$ and $\alpha, \beta, \gamma \geq 0$. The above equation can be solved using linear programming as done in [8].

2.0.2 Architecture of LID systems

A typical architecture of LID system is illustrated in Figure 2.0.1. Each of the components of the LID system is described below:

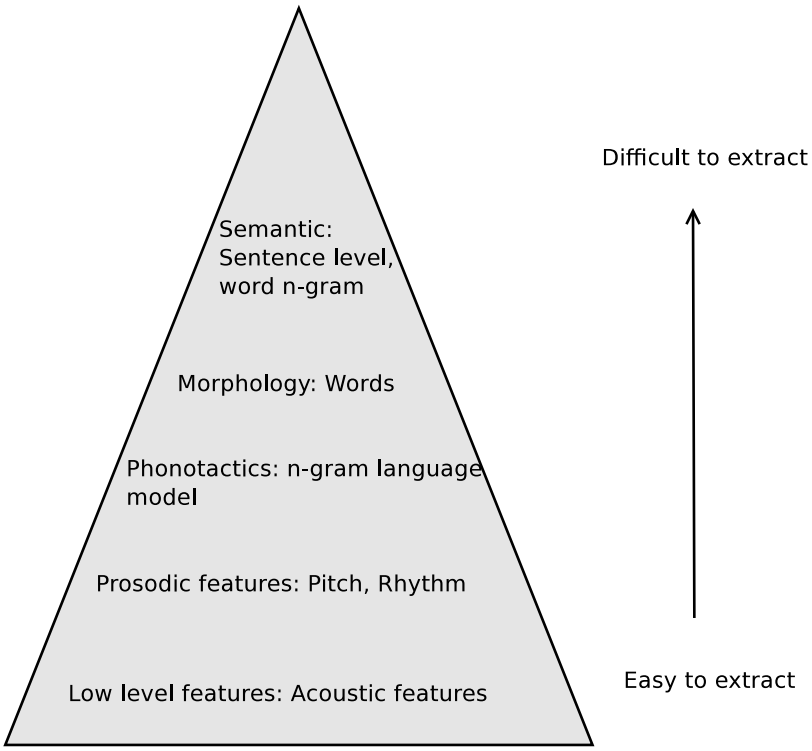

Fig. 2.2: Different levels of feature extraction in LID task

2.1 Tokeniser

The first step is to build a sub-word based tokenizer which is used to tokenize speech signal into phonemes, syllables, etc. The tokenizer can be built by using the speech transcription like phoneme based tokeniser [2] or these subwords can be automatically extracted from the speech signal like

The most successful sub-word unit includes phoneme tokeniser [2], syllable tokenizer [6] and some fuzzy sub-word units [9]. The tokeniser is used to segment the waveform into these subword tokens. As illustrated in Figure 2.2, larger sub word units are more discriminating than smaller units like phones. These following units have been successfully used in LID

- **Frame Level**: The simplest tokenization possible as used in [10] which is followed Gaussian Mixture model to capture the bigram statistics.

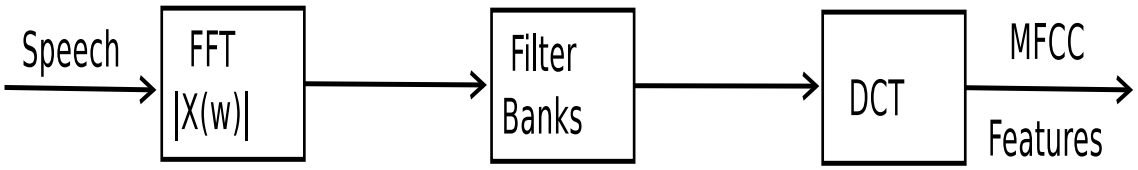

Fig. 2.3: *Extracting MFCC Features from a frame of speech.*

- **Phoneme Level**: The most successful LID system uses phoneme as the sub-word to capture the language characteristics.

2.2 Features for Language Identification

This section describes some of the features that has been used in language identification. It has been shown in [10], by using appropriate shifted delta cepstrum (SDC) features, the error rate decreases by $\approx 8\%$ over the traditional mel frequency cepstrum coefficient (MFCC) features. The important features are described below:

2.2.1 Mel Frequency Cepstrum Coefficient (MFCC)

The MFCC features are used to build state of the art systems in language identification/speaker verification tasks [11] [5]. MFCC features use the fact that human can perceive two low frequency close to each other whereas they can not distinguish two close high frequency component. The block diagram of MFCC feature extraction is shown in Figure 2.3

.

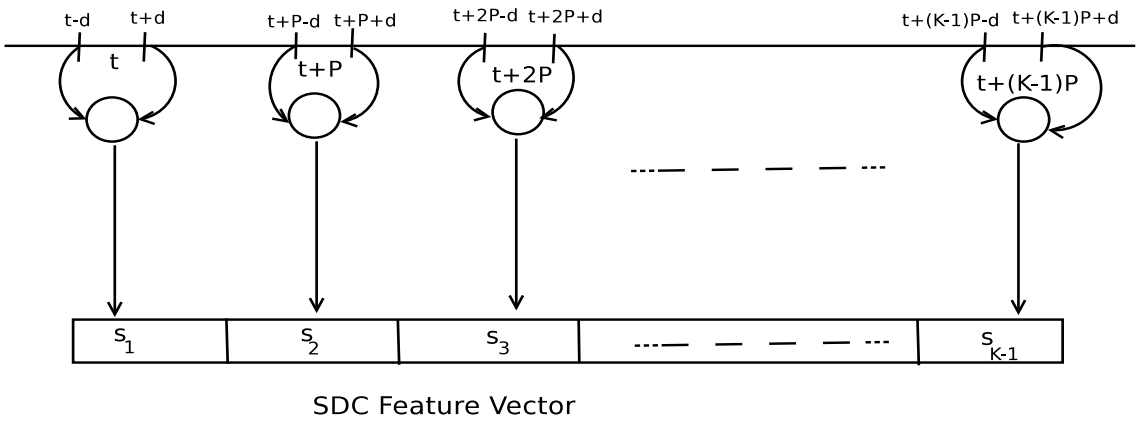

Fig. 2.4: *Extracting of SDC Features from a frame of speech.*

2.2.2 Shifted Delta Cepstrum (SDC)

This is variant of MFCC features and is constructed by stacking cepstra and delta cepstral features. SDC parameters are defined by 4 parameters namely N-d-P-K, where, N is the number of cepstral features, d is time advance, P is the time shift between consecutive blocks and K determines the span of the feature. The feature extraction process is illustrated in Figure 2.4. The dimension of SDC features is kN as compared to 3N dimension MFCC features (N static MFCC features + N velocity MFCC feature + N acceleration features). Hence, from an utterance containing N_u number of cepstral features, we will get only [N_u - (K-1) * P - D] SDC features. The feature at time t is given by the following equation:

$$\Delta c(t) = c(t + iP + d) - c(t + iP - d) \tag{2.8}$$

where 0 $\leq$ i $\leq$ K - 1

2.2.3 Mel Slope

The mel slope is computed as follows:

- Fast Fourier Transform (FFT) of a frame of speech signal (X(ω)).

- Pass the FFT of the signal, log(|X(ω)|) through a bank of filters to get N coefficients.
- The trajectory information of these N coefficients is extracted by taking derivative at each point. The derivative is obtained by least square fit of M values (say 5) to get mel slope of the frame of speech. Hence the dimension of mel slope of a frame of speech is M - 1.

2.2.4 Group Delay Features

Group Delay is defined as the negative derivative of phase as

$$\tau(\omega) = -\frac{d(\theta(\omega))}{d\omega)} \tag{2.9}$$

where θ (ω) is the phase of the frame of speech signal. Equation 2.9 can be simplified to

$$\tau(\omega) = \frac{X_R(\omega)Y_R(\omega) + X_I(\omega)Y_I(\omega)}{|X(\omega)|} \tag{2.10}$$

where the subscripts R and I denote the real and imaginary parts of the Fourier transform. X(ω) and Y(ω) are the Fourier transforms of x(n) and nx(n), respectively. The group delay function requires that the signal be minimum phase or that the poles of the transfer function be well within the unit circle for it to be well behaved. The spiky nature of the group delay spectrum can be overcome by replacing the term $|X(\omega)|^2$ in the denominator of the group delay function with its cepstrally smoothed version as given by equation 2.10. Finally, DCT is done to de correlate the resultant signal.

2.3 Class Model

The class models can be built using mainly two sources of information.

- Acoustic.
- Language Model (LM) or Phonotactics.

2.3.1 Acoustic Model

A simple acoustic model is GMM-UBM system where the front end is GMM. The GMM built by using data of all class and so it represent a general language/class. The gaussians represents the subword acoustic space. The specific language/class model is obtained by pumping data of the class and adapting the parameters of the GMM.

2.3.2 Language Models/Phonotactics

LID systems based on phonotactics uses sequence of phones/sub-word to characterize the acoustic content of a language. It usually consists of building a front end recognizer either using labeled or unlabeled data. LM is used for weighting different sequence of sub words. Suppose a frequent sequence of sub words in a language occurs which does not occur in other language. The LM is supposed to give higher importance or weightage to this particular sequence more than others.

2.4 Evaluation Metric

In OGI MLTS, the average recognition rate for 11 languages was proposed as the evaluation metric. However with the success of NIST-SRE campaign the speaker recognition metrics are now commonly used in language verification context. Performance is mea-

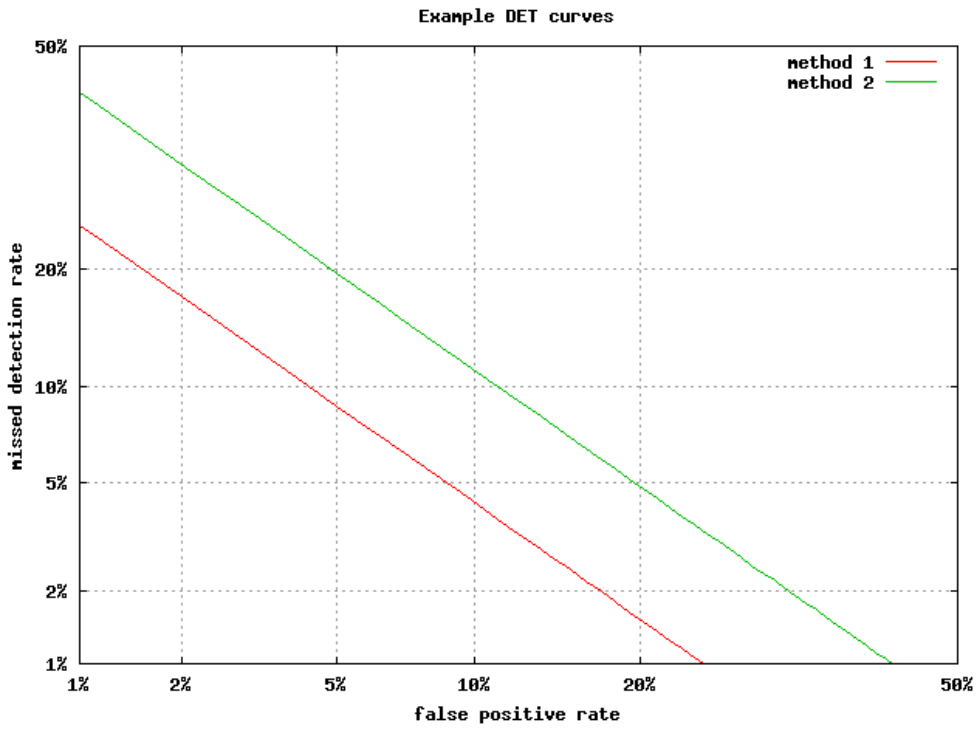

Fig. 2.5: DET Curve

sured by Equal Error Rate which is the point where miss probability and false alarm probability are same.

A detection error trade off (DET) graph is a plot of error rates for binary classification systems, plotting false reject rate vs. false accept rate. The X and Y axes are scaled non-linearly by their standard normal deviates, yielding trade off curves that are more linear than ROC curves. As shown in [12] that if the score distribution in normal then the plot will be a straight line.[2]

2.5 Literature Review

Most of the LID systems have been divided into two major categories (i) phonotactics, and (ii) acoustic based LID systems. Phonototactics based LID system has provided

[2]The axis are not uniformly spaced

superior performance [2]. Some of the LID systems are described in the subsequent sections:

2.5.1 1990-2000

It saw the release of publicly available corpora and most of the works focused on the use of phonotactic information for language identification framework. Some of the notable works includes the following:

- Muthuswamy, 1992:

  In [13] [14], they started a systematic approach to build and evaluate LID systems. The initial experiment was conducted in four languages with high quality speech data. They used neural network to segment seven broad category of speech (namely stop, closure, vowel, silence, sonorant, silence, intervocalic sonorant). From these segments 304 dimension features which included prosodic and acoustic features for classification. They got good classification accuracy which motivated to investigate in larger database with 10 languages. They performed experiments on 10 languages and one vs the rest to test their approach. The best accuracy of the systems were 47.7% on 10 language classification task and 88.6 %. Experiments on humans were also reported where they had to identify segments of speech of duration one, two, three and four second duration out of the 10 target languages. It was found that human use various cues like word spotting strategies (for example words "irnnidu"in Korean), prosodic patterns (for example sing song nature of Arabic language). He concluded that phonetic based LID system works better than system based on broad phonetic categories.

- Zissman, 1996:

In [2], they have compared four approaches to language identification namely Gaussian Mixture Model based system; single language phone tokeniser followed by language modeling of the languages concerned; multiple phone recognizer of the languages and followed by language modeling.

For the GMM based system, they built two separate GMM system one based on cepstral features and other based on delta cepstral features. For recognition, the combined log likelihood is linear in log likelihood of the two systems as following:

$$\log(\boldsymbol{X}_c, \boldsymbol{X}_{dc}|\lambda_c, \lambda_{dc}) = \sum_{t=1}^{T}[\log(\boldsymbol{x}_t^c|\lambda_c) + \log(\boldsymbol{x}_t^{dc}|\lambda_{dc})] \tag{2.11}$$

where

- $\boldsymbol{X}_c = \{\boldsymbol{x}_1^c, \boldsymbol{x}_2^c, \boldsymbol{x}_3^c, \cdots, \boldsymbol{x}_T^c\}$ is the cepstal sequence of feature vector of the test utterance.
- $\boldsymbol{X}_{dc} = \{\boldsymbol{x}_1^{dc}, \boldsymbol{x}_2^c, \boldsymbol{x}_3^c, \cdots, \boldsymbol{x}_T^c\}$ is the delta cepstal sequence of feature vector of the test utterance.
- λ_c and λ_{dc} are the GMM models using cepstral and delta cepstral features respectively.

For the single phone tokeniser followed by language modeling approach, a single phone recogniser trained in one language. The tokenises produces tokenises the training data into language dependent phonemes which is then used to train language model. The statistics are obtained as explained in section 2.3.2.

For the parallel phone recognition followed by language modeling system, multiple phone recognizers are employed to tokenize speech into language dependent phoneme units. These phoneme units are then used for language modeling. The

number of language models in this technique is high. To cite an example if the task at hand is to identify English, Spanish and Hindi and phone tokeniser is available for English and Arabic. The number of language models is 3 * 2 = 6. The best performance is

The parallel phone recognition system is suited when labeled training data is available for training the language model. During the Viterbi decoding, the phoneme boundaries is obtained using the acoustic and phonotactic constrained. The best error rate is 20% on 45 s test and 30% on 10 s test utterance in 11 language classification.

- 1996: Schultz et al. [15] used large vocabulary continuous speech recognition system (LVCSR). They compared language recognition system based on phone level and word level both with and without language model (LM). In the first attempt, bigram LM was implemented, but trigram in the second stage gained better results. On four language task, word-based system with trigram modeling of words (accuracy 84%) outperformed the phone-based system with trigram modeling of phones (82.6%) significantly. They claim: The more knowledge is incorporated in the word-based language recognition system, the better performance.

2.5.2 2000-present

The NIST LRE efforts have resulted in varieties of techniques applied in LID. In this period, the speaker verification techniques have been successfully used in language verification context.

- 2002: Torres-Carrasquillo et. al [10] [16] used GMM as the front end to tokenise training speech data into GMM tokens. Each of the gaussians in Gaussian is

mixture supposed to be a representative of sub-word of the language. These sub-words are smaller than phonemes. These sequence of tokens are then hypothesed to capture the language characteristics similar to the phone based system. Language models based on these tokens was built. Integrating acoustic and language model information was found to give the best performance. They introduce shifted delta cepstral as described in 2.2.2, a variant of traditional SDC features. They found that performance increased by 8 % on using SDC features. While the performance of GMM tokenisation system is worse than parallel phone recognization system. Combining the acoustic likelihood of GMM tokenisation system and PPRLM system the error rate decreased to 6.90 %.

- 2002: Jayram et. al [17] proposed a sub-word based language identification which does not need labeled corpora. They proposed to replace the traditional (i) front end phone tokeniser and (ii) back-end language model with acoustic sub-word model which explicitly captures the phonotactics. The sub-word units are obtained front the speech signal in a maximum likelihood sense such that each of these units should have less intra cluster (sub-word unit) distortion. These sub-word units are then clustered using K-Means algorithm. These each of these M sub-words are thus modeled with a GMM/HMM. Assuming these M sub-words are connected using a HMM [3]. The parameters of the HMM are estimated by decoding the speech into these M sub-words and then counting the big-ram probability value to represent the transition probabilities.

- 2006: Campbell et. al. [18] used Support Vector machine for language classification with SDC features. The main challenge is the design of sequence kernel that measure the distance between two utterances. They constructed kernel

[3]Assuming each of these sub-words can follow any other sub-word.

based on minimum squared error criteria on the high dimension expanded SVM space. They obtained 3.1 % EER in NIST 2003 LRE with the fused system comprising of GMM system and SVM system.

- 2006: Cordoba et. al. [19] [20] suggested improvements to the PPRLM framework for language identification. Firstly, they experimented with fixed and variable threshold in the computation of n-gram phone statistics [4]. They found that using variable threshold scheme reduces the EER considerably by ≈ 35 %. Secondly, they experimented with bias removal strategies [5] at the back-end classifier. They trained Gaussian classifier with the language model score (trigram score) obtained for each of the tokens. It was observed that the EER reduces considerably further by ≈ 14 % with the inclusion of bias removal scheme and using acoustic (phone recognition likelihood) information.

- 2007: Haizhou Li et al. [9] used vector space approach for language classification. They

- 2008: Campbell [21] constructed a covariance SVM kernel. A universal background model (UBM) was built and parameters (mean and covariance matrix) of each of the utterances was adapted [11]. The distance between two utterances utt_1 and utt_2 is given by the expression:

$$K(utt_1, utt_2) \approx \sum_{i=1}^{K} w_i D(N(:, m^i_{utt1}, \Sigma^i_{utt1}), N(:, m^i_{utt2}, \Sigma^i_{utt2})) \tag{2.12}$$

  where

  - K is the number of mixture component of UBM, w_i is the weight of i^{th} mixture.

[4] It is similar to language model smoothing techniques

[5] Biases are introduced at the language model scores

- m^i_{utt1} and m^i_{utt2} are the adapted means of the i^{th} mixture of the utterances 1 and 2.
- Σ^i_{utt1} and Σ^i_{utt2} are the adapted covariances of the i^{th} mixture of the utterances 1 and 2.
- $D(N(:, m^i_{utt1}, \Sigma^i_{utt1}), N(:, m^i_{utt2}, \Sigma^i_{utt2}))$ is the Kullback Leiblar divergence between two Gaussian distributions [22].

They obtained an EER of 3.2 % on NIST 2005 LRE and in the 30 s duration test.

- 2010: Rong Tong et. al. [4] used feature selection in the PPRLM framework. In a traditional system, each of the front-end phone recognizers will tokenise the speech into sequence of tokens. The n-grams are based on the statistics of these tokens. They used SVM and χ^2 test to measure the relevance of these features/n-gram statistics in high dimensional feature space. As removing the feature which has less count is not a good idea as we have have see the distinguishing characteristics as well. They found that SVM based feature selection mechanism works better than χ^2 test when the dimension is reduced considerably. They equal error rates (EERs) of 1.84 with 4-gram statistics on the 2007 NIST Language Recognition Evaluation 30s closed test sets.
- 2011: Alberti et. al. used discriminatingly trained features for language identification. The features are assumed to have obtained by the equation:

$$x_u(t) = o_u(t) + Mh_u(t) \tag{2.13}$$

where $o_u(t)$ is the feature vector and $h_u(t)$ is a sparse vector. Each of the utterances are adapted from the UBM using the equation 2.13. The means of the utterances for each of the mixtures are stacked to make a super-vector. They

proposed to solve M in an iterative manner as: (i) solve the SVM dual form to obtain the optimal value of the support vector weights α with M fixed (intialise M as zero matrix) and (ii) move M in a direction of anti gradient as done in gradient descent algorithm.

They experimented in NIST 2003 LRE and EER decreased by 5.9 %.

The state of the art LID system uses parallel phone recognition followed by language modeling (PPRLM) to capture the characteristics of a language. This system is referred to as the explicit LID system which needs labeled speech corpora. An alternative scalable approach is the implicit LID system which does not need annotated speech database. A popular implicit LID system is the Gaussian Mixture Model (GMM) tokenizer as described in [10]. It uses GMMs as the front end to tokenize an incoming speech utterance into cluster indices. These cluster indices are then used to create interpolated Language Models that discriminate among languages. The performance of this system is comparable to Parallel PPRLM system on the OGI-MLTS database.

The sub-word unit based LID system has consistently given good performance as evidenced from the NIST evaluations. Researchers have preferred to use phonemes over other sub-word units like syllables. Phonemes by itself cannot identify a language as languages share the same phoneme inventory. However sequence of phonemes contain the distinguishing characteristics and it has been found that trigram or higher n-gram phonemes statistics gives higher accuracy [4]. A syllable on an average contains three phoneme units. We hypothesize that syllables inherently contain distinguishing characteristics. To cite an example on the importance of syllables, if the domain of discourse is Indian languages, the presence of the syllable */zha/*, reduces the search space to two languages, namely Tamil and Malayalam. The other advantage of using

syllables as a sub-word for building an LID system is that it can be automatically extracted from the speech signal.

Nagarajan [23] was the first to suggest the use of syllables for building implicit LID systems. The drawback of the approach used by Nagarajan is that the parameters of the incremental HMM based clustering approach, significant tuning of parameters during training [6]. This is primarily because syllable models built from insufficient data suffer from modelling errors. Iterative tuning of parameters is required to ensure that robust syllable models are built [6]. To address this problem, we have explored two different approaches: i) top down approach approach to cluster syllables, and ii) a universal syllable model framework. While the performance [6] of the two proposed syllable approaches are worse than that of the baseline for OGI-MLTS corpus, it performs better than the baseline system for NIST 2003 LRE database. Although the performance is not comparable to that of state-of-the-art systems (3% EER for the NIST 2003 LRE database), it is important to note that we have only used the OGI-MLTS and NIST 2003 training and development corpora for building the models [7].

2.5.3 Baseline segmentation algorithm

A syllable is made up of an *onset, rime and coda*. The onset and coda consist of one or more consonants, while the rime is a vowel. If the short-term energy is used to characterize a syllable, the vowel region has much higher energy compared to that of the onset and coda. A time-domain acoustic segmentation approach was proposed in [24] which uses the property that every syllable will have peak energy at the vowel. But the short-term energy function (STE) can have problems of local energy fluctuations.

[6] Performance can be measured by identification accuracy, Equal Error Rate.

[7] State of the art systems use Callfriend Database for Training Models.

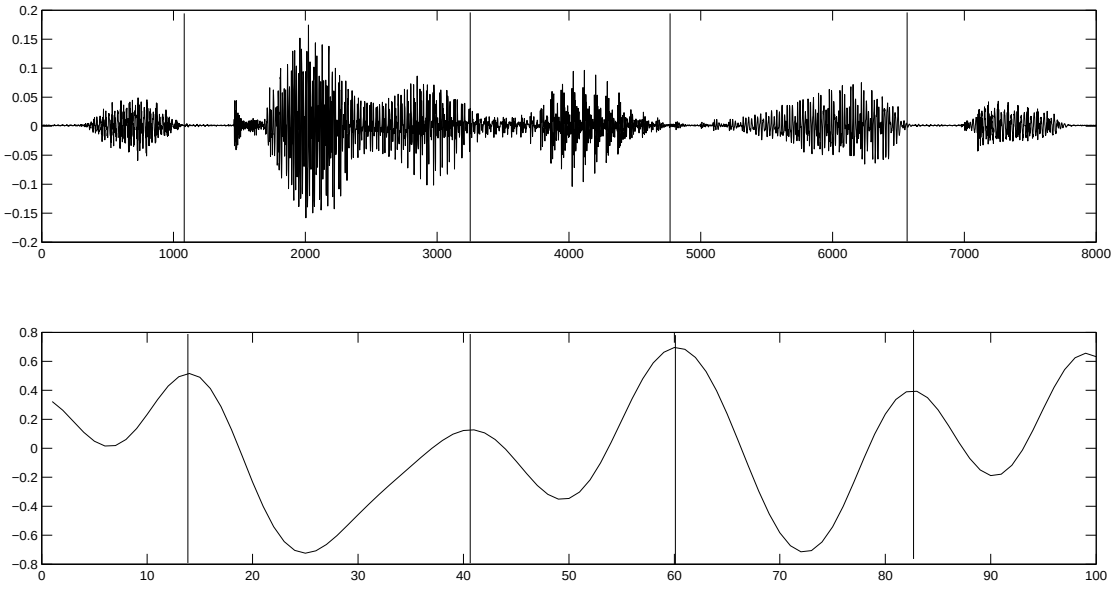

Fig. 2.6: *Segmentation of an utterance taken from OGI Database. The peaks in the second plot gives the location of the syllable boundaries.*

To smoothen the STE, a group delay approach is used.

Using the baseline segmentation algorithm, all the training speech data of each language are segmented into syllable-like units resulting in M_l syllable like units for each language. These syllable like units are used during clustering process to obtain representative syllable models of the language.

2.5.4 Baseline Syllable Clustering Algorithm

A syllable clustering algorithm has been proposed in [23] to cluster similar sounding syllables. From these similar sounding syllables, models for each of the languages is created.

The syllable clustering algorithm is divided into two phases as:

- Initial Cluster Selection: To initialize the parameters of the syllable models.

Each of the syllable models are created from just one syllable instance with multiple frame rate.

- Incremental Training : Derive representative model for each of the language.

2.5.4.1 Initial Cluster Selection

A subset of N (N $\leq$ M) syllables out of the M syllable of a language is selected to obtain an initial set of clusters. The algorithm is described below:

1. Features (13 MFCC + 13 Δ + 13 Δ Δ) are extracted from each syllables. Features are extracted from each syllable instances using multiple frame rate and multiple frame size.
2. N hidden markov models (HMM) models are created corresponding to each of these N syllables.
3. These N syllables are then decoded using these N HMM models created in the above step using 2 best recognition criteria. It results in N syllable pairs. Syllables which are repeated in more than one clusters syllable pair, one of the cluster is removed and number of clusters is reduced.
4. Create new syllable model by re-estimating the parameters
5. Iterate the above two steps twice.

2.5.4.2 Incremental Clustering

In this phase, the final clusters representing the various sound units are obtained.

The algorithms is described below

1. The syllable models obtained in the Initial clustering phase as described in section 4.1 is used to decode the M syllables.

2. Clustering is done based on the decoded output. If a cluster is found to contain less than 5 syllables then the cluster is removed.

3. Re-estimate the parameters of the clusters after this step.

4. Repeat the above steps 5 times.

The clustering process gives $'C_l'$ representative syllable models for each language.

2.5.5 Testing

Various methods like acoustic likelihood, voting etc can be used to evaluate the performance as described in [23]. Acoustic log-likelihood scores is used as the criteria for evaluating system as it gives the best performance [23]. The scoring method is described below:

- An utterance is segmented into syllable like units (s_1, s_2, $\cdots$, s_N) and each of these syllables is scored against the model of each of the languages. For each of the languages,

  $p(s_j|\lambda_i) = \max\ [p(s_k|\ \lambda_i)]$

  where $1 \leq$ k $\leq$ N and s_l is the highest scoring syllable.

- Accumulate the log likelihood scores and pick the the language ($l^{'}$) that gives the maximum score.

$$l^{'} = \underset{l}{argmax} \sum_{k} logp(s_l|\lambda_i) \tag{2.14}$$

CHAPTER 3

Unsupervised clustering of syllables for Language Identification

In this section the drawbacks of the baseline system is discussed and two approaches to cluster syllables are explored, namely, (i) top down clustering approach, and (ii) Universal syllable model.

3.0.6 Drawback of the baseline clustering approach

The baseline syllable clustering algorithm approach is a bottom up approach (agglomerative clustering) where each of the syllable models (HMM's) is built with only one syllable in the initial stages of the algorithm. In subsequent iteration, similar sounding syllables are grouped to form a cluster. Distance between two syllables is determined by scoring it against all the models and picking the closest match. At termination of the algorithm, these clusters consists of sufficient number of syllables. The stopping criteria used in this algorithm can be (i) desired number of clusters or (ii) clusters does not move [6]. It is to be noted that this approach has the problem that syllable models may not be built in the initial phase of the algorithm and this is inspite of multiple frame rate and multiple window size [6]. An iterative tuning of parameters is required to ensure that models have sufficient number of syllables. An alternate approach is to use divisive clustering approach where initially one syllable model is built with data

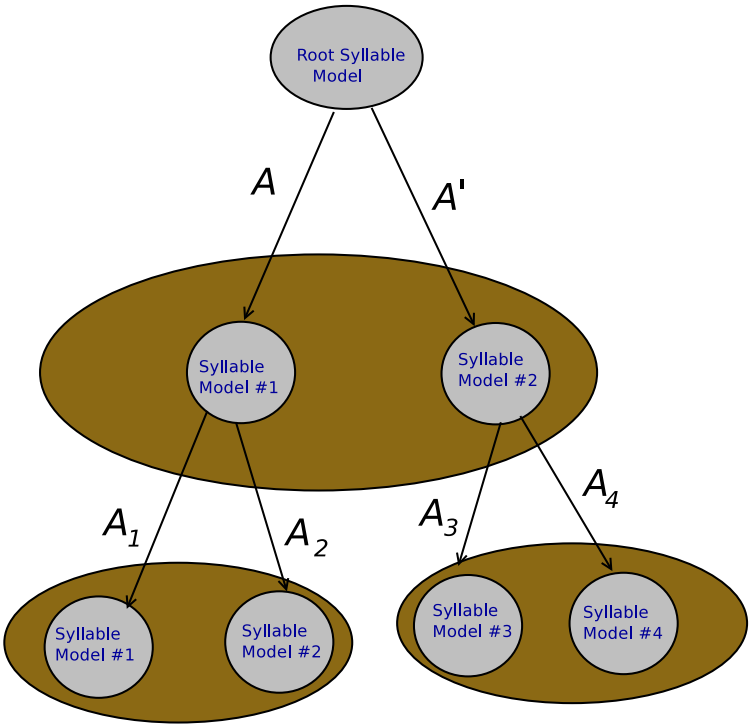

Fig. 3.1: Obtaining syllable models by top down clustering.

from all the syllables [25]. In later stages of the algorithm, each of the syllable models (HMM) is split subsequently to obtain two models. The stopping criteria used is the same as agglomerative clustering.

The main problem with divisive clustering is that in general case the time complexity is $\mathcal{O}\ (2^n)$ compared to $\mathcal{O}\ (n^3)$ of agglomerative clustering approach. It makes divisive clustering less attractive for large data set. However, some heuristic can be made to make it faster and has been described in the next section.

3.1 Top Down Clustering

The data insufficiency problem can be addressed by building syllable models in top down fashion and requiring each of the models thus formed in each level to have required/sufficient number of examples. As illustrated in Figure 3.1, the root syllable

model (λ_1^1) is created with sufficient amount of syllables (N_s). The parameters of the root model can be transformed by matrices A and $A^{'}$ to create two separate models (λ_2^1 and λ_2^2) at level 2. The parameters of the new models can be recomputed by scoring the syllable against the models and re-estimate the statistics using these new information. Applying the process repeatedly to obtain appropriate number of syllable models (λ_l^1, λ_l^2, $\cdots$, $\lambda_l^{l_n}$) at level l. Note that it is assumed that cluster purity [1] of the models at the leaves of the tree is higher than at any level of the tree.

3.1.1 Computation simplification

To address the problem of data insufficiency to build models, it is that an syllable model on an average can be built with atleast N_s (say 30) syllables. N_s can be set (i) manually, or (ii) depend on the amount of data ($N_s^{'}$), or (iii) combination of both like $N_s = \max(30, N_s^{'})$. It is also ensured that at level i, only i * N_s syllables are used for scoring against the existing syllable model as opposed to taking all the syllables. It ensures computation speed up required to score the syllables and build models from them.

In the absence of labeled data, computing these matrices is a big problem and hence it is assumed that all the matrices at all the levels are the same i.e. $A = A^{'} = A_1 = A_2 = ... = A_L$ To further simplify the computation, only the means of the parent HMM's nodes are transformed and updated; and means of the two child nodes are given by the equation $\mu_p \pm \sigma_p$ where μ_p, σ_p is the mean and variance of the parent HMM models respectively [2]. The algorithm is described in detail in the following subsection.

[1] Cluster purity is measured by similar sounding syllables at a given cluster

[2] The Gaussian of HMM's of each state are assumed to have diagonal covariance matrix

3.1.2 Algorithm to built syllable models in top down fashion

The advantage of this process is that (a) number of clusters can be controlled, (b) training time is faster and (c) pruning process is very simple and there is no problem of data insufficiency at any stage. The algorithm is described below:

Let us define $N_s = N/M_\theta$ be the effective number of syllables per model, M_i is the number of syllable model at i^{th} iteration ; M_θ is the number of models expected; $\mu^{(x,j)}$ and $\sigma^{(x,j)}$ are the mean and variance parameter of the x^{th} syllable model's j^{th} state. (Note : here we assume that each state of the HMM has only 1 mixture)

- Initialise the algorithm with a root syllable model (λ_1^1) obtained from N_s syllables.

For *ith* iteration:

1. Create 2 syllable models from each syllable model (λ_i^x and λ_i^{x+1}) with means $\mu^{(x,j)} + k\ \sigma^{(x,j)}$ and $\mu^{(x,j)}$ - $k\ \sigma^{(x,j)}$ respectively from these M_i models.
2. Score i*N_s syllables against 2*M_i models and assign the index of the highest scoring syllable model.
3. Discard the models having less than $N_{s,min}$ syllables and the models after this step be $M_i^{'} \leq 2{*}M_i$
4. Re-estimate the parameters of these $M_i^{'}$ models and $M_i = M_i^{'}$

Repeat Step $1-4$ until $M_i \leq M_\theta$.

The algorithm converges when $N_s > N_{s,min}$ and requiring $N_{s,min} > 10$ and $N_s > 20$ to ensure that models have sufficient number of examples.

3.2 Universal Syllable Model (USM)

The advantage of a large mixture GMM/HMM is that it can capture the variability of the acoustic space and it has been shown that large mixture GMM performs better than fewer mixture model. However, building a large mixture model needs large amount of data for each of the classes. It is to be noted that building a large model using less data leads to parameters of the models not estimated correctly and necessarily results in poor performance. The trade off between large mixture model and amount of data is solved in the context of Speaker Verification literature by using background model. This idea can also be used in syllable based framework for LID system. The background modeling approach to language identification consists of these steps:

- pool data from all the classes and build a single Gaussian Mixture Model (GMM) which is called Universal Background Model (UBM).
- derive a class model $'l'$ by adapting the parameters (only means μ_i^l are adapted) of the UBM using that class training data as given by the following equation:

$$\mu_i^l = \alpha E_i(x) + (1-\alpha)\mu_i^{ubm} \tag{3.1}$$

where,

- μ_i^{ubm} is the mean of the UBM,
- $E_i(x)$ is the new estimate of mean and
- α is the adaptation coefficient controlling the balance between the old and the new mean.

In this thesis, HMM adaptation framework similar to GMM-UBM framework is explored. Furthermore, possibility of using as single and multiple universal syllable model

have been explored in remainder of the section.

3.2.1 Single Universal Syllable Model (Single USM)

The motivation for this approach is that the most of the syllables in all languages have low energy region (the consonant part) followed by high energy region (the vowel portion) followed by low energy region (final consonant region). The theory is reinforced with Figure where the KLD between consecutive frames of syllables is plotted. Is is evident from the figure that the number of transitions in syllables can be taken as constant ($\approx$ 5 state HMM).

In this approach we assume that the syllable models in all the languages can be derived from a global syllable model by adapting the means of the mixtures in every state. The algorithm to obtain each class model is described below:

- A single HMM model is initialized by pooling N syllables from all the languages.This model is referred to as universal syllable model.
- The syllable models obtained in the baseline initial clustering phase and incremental clustering phase are derived from this universal syllable model.

This algorithm overcomes the problem of data insufficiency in the baseline initial clustering selection phase. However using single global syllable model may lead to over generalization as syllable structure of languages are different. We relax the assumption of a global syllable model to introduce a set of universal syllable models.

3.2.2 Multiple Universal Syllable Model (Multiple USM)

The major steps in this approach as shown in Figure 3.3 are creating the universal syllable set and deriving models for each of the languages. The method to create the

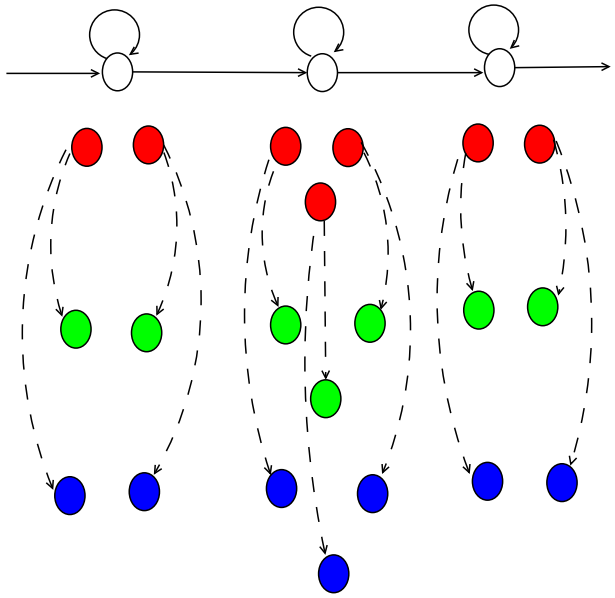

Fig. 3.2: Multiple Universal Syllable Model Approach: Red ellipses represent constant density curves for the GMMs of universal syllable model, green and blue ellipses correspond to GMM of English and Farsi language syllable models respectively obtained after adaptation.

same is described below

- Randomly select N syllables from each of the languages to form an universal syllable inventory (s_1, s_2, s_3, $\cdots$, s_N). These syllables are then clustered using the baseline syllable clustering algorithm. The clustering process will result in 'L' clusters which we refer to as universal syllable model set.

Models for each of the languages is derived from the universal syllable models as follows

- The training utterances of every language are segmented into syllables. Each syllable is scored against the universal syllable model and then it is assigned to the highest scoring syllable model.
- adapt the means of GMMs in every state using the result of previous step. As shown in Figure 3.2, the gaussian of each state english syllable model (green ellipse) or Farsi syllable model (blue syllable) is moved to new position in the acoustic space. It is to be noted that the gaussians of each state in the universal syllable models have one to one correspondence with the gaussians of each of the languages syllable models.

The main advantage of this method is that it reduces the training time by $\approx 10\%$ of that of a baseline syllable based system.

3.3 Experiments

In this section, the results of the baseline syllable and universal syllable based LID systems are described. For OGI database, 40 utterances of 45 seconds duration each are used for training, 20 utterances of 45 seconds each are used for testing and development. For NIST 2003 LRE, only the 80 utterances each of 30 seconds duration from every

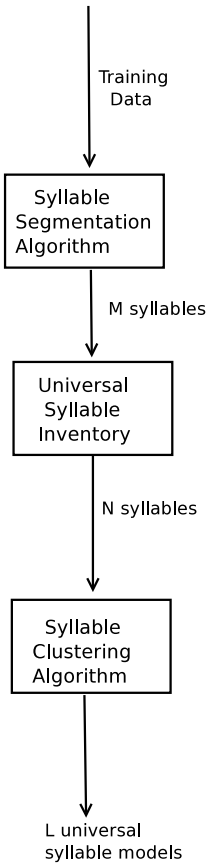

Fig. 3.3: Obtaining Universal Syllable models from training data

language corresponding to primary condition is considered as evaluation data. Since CallFriend database is not avaliable, therefore OGI-MLTS and 1996 development and evaluation data is used as training data for NIST 2003 LRE.

Closed set language identification was performed on OGI-MLTS [26] and NIST 2003 LRE [27] databases. The language identification results are shown in Table 3.1.

Table 3.1: *Performance of LID systems on OGI-MLTS and NIST 2003 LRE.*

LID System	OGI-MLTS	NIST 2003 LRE
GMM-UBM	45 %	35%
Baseline System	62.72 %	39.40 %
Single USM	53.60 %	38.20 %
Multiple USM	55 %	43.54 %
Multiple USM Training and Dev	63.63 %	-
Top Down Clustering	58.63 %	**45.62** %
Top Down Clustering Training And Dev	**68.18** %	-

3.3.1 Baseline Syllable based LID system

For OGI-MLTS database, models were built from 5000 syllables from each of the languages. For each of the languages, 370-390 representative models were obtained after the clustering process. It was observed that the best performance is **62.72 %** when each of the syllables is created with 5 state, 1 mixture HMM. It was also observed that increasing the syllables inventory size does not increase performance and 5000 syllables is optimal.

Due to the limited amount of training data in 2003 NIST LRE, **39.4 %** accuracy was obtained. Since development data was not available for optimizing parameters, the parameter that is obtained with the best performance of OGI-MLTS data-set was used.

Syllable based LID system is compared with state of the art GMM-UBM system using shifted delta cepstral features as described in [5]. The SDC feature extracted with the configuration of 7-1-3-7 to obtain 56 dimension feature vector. Identification accuracies of **45.9%** and **35.65 %** were obtained in OGI-MLTS and 2003 NIST-LRE database respectively. It can be concluded that syllable based LID system works better than SDC based GMM-UBM system with limited amount of data.

3.3.2 Top Down Clustering

The root syllable model is created with 5 mixture and 1 state HMM. The optimal value of k is found to be 0.01 and it is obtained by optimising the performance on the development data. Another method of determining k would be to take some male and female data and find the value of k which divides the male and syllable data into two groups. In OGI database as the syllable inventory size is increased, the best performance is obtained with $N_{syll} = 128$ and $N_s \approx 40$ examples. It was further observed that when the data is increased by pooling the training and development data the performance increased by ≈ 10 % with $N_{syll} = 256$ and $N_s \approx 33$ examples. The performance is attributed to the fact that building more syllable models can better model the language.

In 2003 NIST database, 1996 development and evaluation data were used for building the syllable models. The performance of the system for different cluster size N_{syll} is shown in Figure 3.1. The best performance is obtained with $N_{syll} = 512$ clusters

and $N_s \approx 30$ examples. The performance degrades as we increase N_{syll} to 1024 and when $N_{syll} = 128$ and 256 clusters. The reason of this behavior is that vocabulary of NIST 2003 LRE is larger compared to OGI database as evident from the result.

3.3.3 Single Universal Syllable Model

Since there is no scarcity of data to create the universal syllable model, different HMM parameters were used for optimising parameter in the development set. It was observed that in OGI-MLTS database, 5 state and 3 mixture performs the best, the accuracy being **53.6 %**.

In 2003 NIST LRE, the parameters are configured as obtained with OGI database. The accuracy is **38.2 %** .

3.3.4 Multiple Universal Syllable Model

500 syllables from each language to form the Universal set of syllables and each syllable is represented by 5 states and 1 mixture HMM. These models are then clustered as in the baseline system. The clustering process results in 445 syllable models. These syllable models are then adapted for every language. After adaptation, 395-405 syllable models were obtained for each of the languages. For OGI-MLTS corpora, the language recognition performance is **55 %**. To evaluate the performance of multiple universal syllable models on amount of training data, training and development data were used for the adaptation process and accuracy of **63.63 %** (see Table 3.1, column 4) was obtained.

For NIST 2003 LRE, 1996 development and evaluation data were used for adaptation process to obtain models for every language. English syllable model trained using the OGI-MLTS corpus was used with the baseline syllable LID system as the univer-

sal syllable models.These models are adapted for every language. The performance is found to be **43.2 %**. To evaluate the effect of choosing universal syllable models we used the following language model trained in OGI-MLTS obtained by the baseline system as the universal syllable models: Farsi syllable model and French syllable model. We found that these syllable models the average performance is almost the same.

3.3.5 Discussion

For the OGI database, when training and development data were used to build the models, the performance of the base line system decreased. The reason for this behavior is that the parameters of the baseline system were optimized for this database. The number of parameters in the universal syllable models approach is the same as the baseline syllable system with the exception of relevance factor which is set to default value of 10.

3.4 Conclusion

In this chapter, top down clustering approach and universal syllable model approach to language identification are explored in this paper. It was observed that top down clustering approach performs better than universal syllable model approach in OGI-MLTS and NIST-2003 databases. Nevertheless, training time for universal syllable approach is significantly less than than of the baseline system and top down clustering.

CHAPTER 4

Spoken Language Identification using Phonotactics

4.1 Phonotactics

"Phonotactics is a branch of phonology that deals with restrictions in a language on the permissible combinations of phonemes. Phonotactics defines permissible syllable structure, consonant clusters, and vowel sequences by means of phonotactical constraint. Phonotactic constraints are language specific" (source:: wikipedia) For example all the words in Telugu language have vowel endings and consonant cluster like /st/ does not occur in Japanese language.

4.2 Probabilistic framework for incorporating phonotactics information

Suppose a database consists of the utterances $\boldsymbol{u_1}$, $\boldsymbol{u_2}$, $\cdots$, $\boldsymbol{u_N}$. Each of these utterances consists of the tokens $\boldsymbol{u_i}$ = ($\boldsymbol{a_1^i}$, $\boldsymbol{a_2^i}$, $\cdots$, $\boldsymbol{a_k^i}$). Hence the problem of language identification is that given an test utterance $\boldsymbol{u_t}$ = ($\boldsymbol{a_1}$, $\boldsymbol{a_2}$, $\cdots$, $\boldsymbol{a_k}$) can be mathematically formulated as:

$$p(a_1, a_2, a_3, \cdots, a_N) = \prod_t p(a_t | a_{t-1}, \cdots, a_{t-M+1}) \tag{4.1}$$

for M gram statistics. These statistics can be computed from the corpora using the following equation:

$$p(a_t | a_1, \cdots, a_{t-M+1}) = \frac{p(a_t, a_{t-1}, \cdots, a_{t-M+1})}{p(a_{t-1}, \cdots, a_{t-M+1})} \tag{4.2}$$

These statistics can be computed from the corpora using the following equation:

$$p(a_t | a_1, \cdots, a_{t-M+1}) = \frac{\#(a_t, a_{t-1}, \cdots, a_{t-M+1})}{\#(a_{t-1}, \cdots, a_{t-M+1})} \tag{4.3}$$

where # (a_t, a_{t-1}, $\cdots$, a_1) is the count of the subword sequence a_t, a_{t-1}, $\cdots$, a_1. One of the problem of obtaining frequency values in this way is that some of the n-gram may not occur in the training data and will be assigned zero probability. To dispose of this problem, some form of smoothing is done such that the probability mass function sum to one. Some successful smoothing techniques include add one , Good Turing discounting, back-off smoothing techniques.

As evident from NIST evaluations [**?**], the most successful technique of language identification is to use acoustic information with language model (LM). The choice of phones to model the characteristics of a language is shown to perform the best. In [**?**], it is shown that accurate phone recognizer gives higher language identification accuracy. But building phone recognizers require annotated speech corpora which makes it less attractive to build LID systems. These systems are called Explicit LID systems which need labeled database. An alternative is to built Implicit LID systems which do not need labeled database. One such implicit LID system uses Gaussian Mixture Model (GMM) as the front-end tokeniser and a set of language models. The advantage of this approach is that even in a small database it is possible to obtain a reliable bigram statistics that distinguishes among languages. Another implicit LID

system is syllable based LID system as described in [6]. The syllable based LID system automatically segments speech signals into syllable like units and then builds representative models for every language. It is observed that acoustic score is enough to distinguish among languages. In this chapter, the performance of syllable based LID system and GMM tokeniser based system are compared in single/multiple universal model(s) paradigm. Both acoustic score based LID system and language model based system are compared. GMM LID system as described in [10] is used as the baseline system. However in syllable based systems, universal syllable models as described in the previous chapter is used to derive the language model for each of the classes.

As illustrated in Figure 4.1, the two main components of the LID system considered in this paper, namely, universal tokeniser/model and class models. In acoustic based system, each of the class models are adapted from this universal model whereas in language model (LM) based LID systems, each of the class LM are derived from the tokens of the universal model.

4.3 Baseline GMM Tokeniser using SDC features

In [10], LID system based on GMM tokenisation using Shifted Delta Cepstral features provided good performance. Hence this system is used as the baseline LID system. SDC features are obtained by stacking delta mel frequency ceptral coefficient (MFCC) across multiple frames. The features are defined by four parameters N-d-P-K, where, N is the number of MFCC features, d is time advance, P is the time shift between consecutive blocks and K determines the span of the feature. The details of SDC features can be found in [10] and [5].

In the baseline system, a front end GMM tokeniser is used to tokenise speech

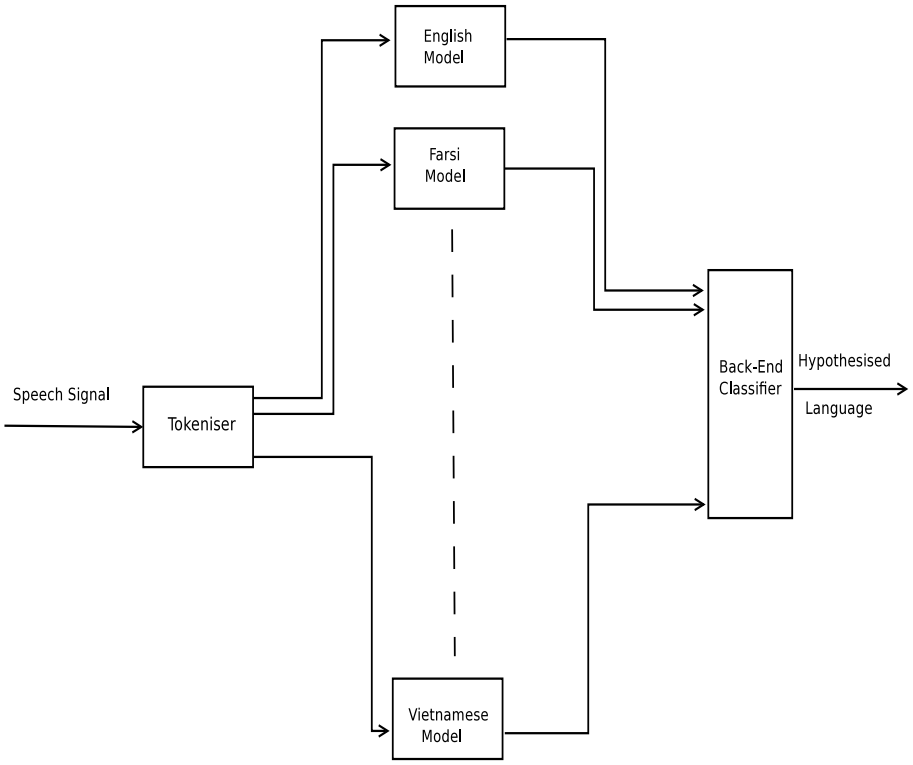

Fig. 4.1: *General structure of LID system.*

signal into GMM component indices. Interpolated language models for each of the classes are built on these indices to capture the bi-gram statistics of the cluster indices. Interpolated language model can be expressed by the equation,

$$P^{'}(a_i, a_j) = c_2 P(a_i, a_j) + c_1 P(a_i) + c_0$$

where, $P(a_i, a_j)$ is the bi-gram probability of tokens a_i and a_j; $P(a_i)$ is the uni-gram probability of the token a_i; c_2, c_1 and c_0 are the bi-gram, uni-gram and offset coefficients respectively.

The acoustic models for each of the classes are derived from the universal syllable models by Maximum a Posteriori (MAP) [11] adaptation. The universal syllable models are used to derive language models for each of the classes. The details are given in next section.

4.4 Language Identification using Language Model

Phone recognition followed by language modeling is one of the successful LID systems in NIST evaluations. In [4], it is claimed that using 4-gram phones statistics provides better performance than 2-gram or 3-gram phone based LID system. On an average a syllable contains 3 phones which makes it an excellent candidate for language identification. It is expected language modeling based on syllable like token will provide good performance. The main advantage of syllable based approach is that syllables can be automatically extracted from the speech signal unlike phones which needs transcription.

The main components of this approach are

- Obtain N-gram syllable statistics, and
- Testing: classifying an utterance.

4.4.1 Obtaining N-gram statistics

Obtaining N-gram statistics of a language 'l' from a set of utterances is described below :

- Tokenize each of the utterances into syllable like units and score each of the syllables against the universal syllable models. Assign the index to the highest scoring syllable model.
- Obtain uni-gram/bi-gram syllable statistics using these indices produced in the previous step.

4.4.2 Testing Phase

For a test utterance, similar to training phase, tokenize it into syllables and obtain the N-gram statistics which is represented by the vector $\boldsymbol{H_{tst}}$. To classify an utterance, cosine distance is used and is defined as

$$D(\boldsymbol{H_{l'}}, \boldsymbol{H_{tst}}) = \frac{\boldsymbol{H_{l'}^T H_{tst}}}{|\boldsymbol{H_{l'}}||\boldsymbol{H_{tst}}|} \tag{4.4}$$

and choose the language 'l' for which cosine distance is maximum and is formally described as

$$l = \underset{l}{argmax}[D(\boldsymbol{H_{l'}}, \boldsymbol{H_{tst}})] \tag{4.5}$$

where $\boldsymbol{H_{l'}}$ represents the vector containing N-gram statistics of language 'l''.

Two LID approaches, namely, histogram and vector space based approach are described that use the above methods but in a different way.

4.4.3 Histogram approach

In the training phase, a global N-gram syllable statistics is obtained from the training data of language 'l' which is represented by the vector $\boldsymbol{H_l}$ (or histogram). Classify an utterance using the cosine distance using equation 4.5.

4.4.4 Vector Space approach

The basic idea in this approach is to view an utterance like a text document [9]. Each of the utterances of a language is represented by a vector,

$$\mathbf{v} = \begin{bmatrix} w_1 & w_2 & .... & w_N \end{bmatrix}^T$$

where, w_i is specific weight consisting of local and global probability value of i^{th} syllable. Mathematically w_i is expressed as:

$$w_{t,d} = \mathrm{tf_{t,d}} \cdot \log \frac{|\mathrm{D}|}{|\{\mathrm{d}' \in \mathrm{D} \,|\, \mathrm{t} \in \mathrm{d}'\}|}$$

- $\mathrm{tf_{t,d}}$ is term frequency of term t in document d (a local parameter) and
- $\log \frac{|D|}{|\{d' \in D \,|\, t \in d'\}|}$ is inverse document frequency (a global parameter). $|D|$ is the total number of documents in the document set; $|\{d' \in D \,|\, t \in d'\}|$ is the number of documents containing the term $''t''$.

The global probability value is obtained as explained in section 4.4.1 and local probability value is obtained by considering the N-gram syllable statistics of an utterance. In this approach, 'M' such vectors for a language 'l' corresponding to each of the training utterance are obtained. K-Nearest Neighbour (K-NN) classifier is used to classify a test utterance.

For building the baseline GMM tokeniser system, histogram and vector space based approach are used and are described above except that the tokens are obtained at frame level.

4.5 Multiple Tokeniser Based Approach

As illustrated in Figure 4.2, this approach consist of:

- ***Tokenizer***: As shown in [2] [10] that performance of LID system increases as number of tokenizers are increased. The state of the art LID system phone

recognition followed by language modeling [2]. uses 6 phone tokenizer to decode incoming speech data.

- ***LanguageModel***: In the universal syllable LID system, N language needs to be built corresponding to N languages. However in this case, N * N language models needs to be built with N language models corresponding to 1 tokenizer.
- ***BackEndClassifier***: This module plays an important part. As reported in the literature, biases in the language model score are observed. One of the reasons of this biases are that the tokenizers have different levels of granularity.

4.6 Experiments

In this section, the results of baseline GMM-UBM system and universal syllable based LID systems are described. OGI-MLTS [26] is used to perform closed set language identification experiments and training, development and test data used are as described in the experimental section of previous chapter.

4.6.1 Baseline GMM-UBM system using SDC features

The shifted delta cepstral features are configured as the successful configuration in LID literature of 7-1-3-7 (N-d-P-K). Hence, 56 dimension feature vector is obtained for each of the frames consisting of 49 stacked delta coefficient + 7 static MFCC features. The features are mean removed and variance normalized per utterance basis.

Five utterances from all the languages are taken to build UBM. Due to limited amount of data we have experimented with 64, 128 component GMM. The acoustic models for each of the classes are adapted from the UBM. It was observed that using 128 component GMM gives the optimal performance of **44.5** %.

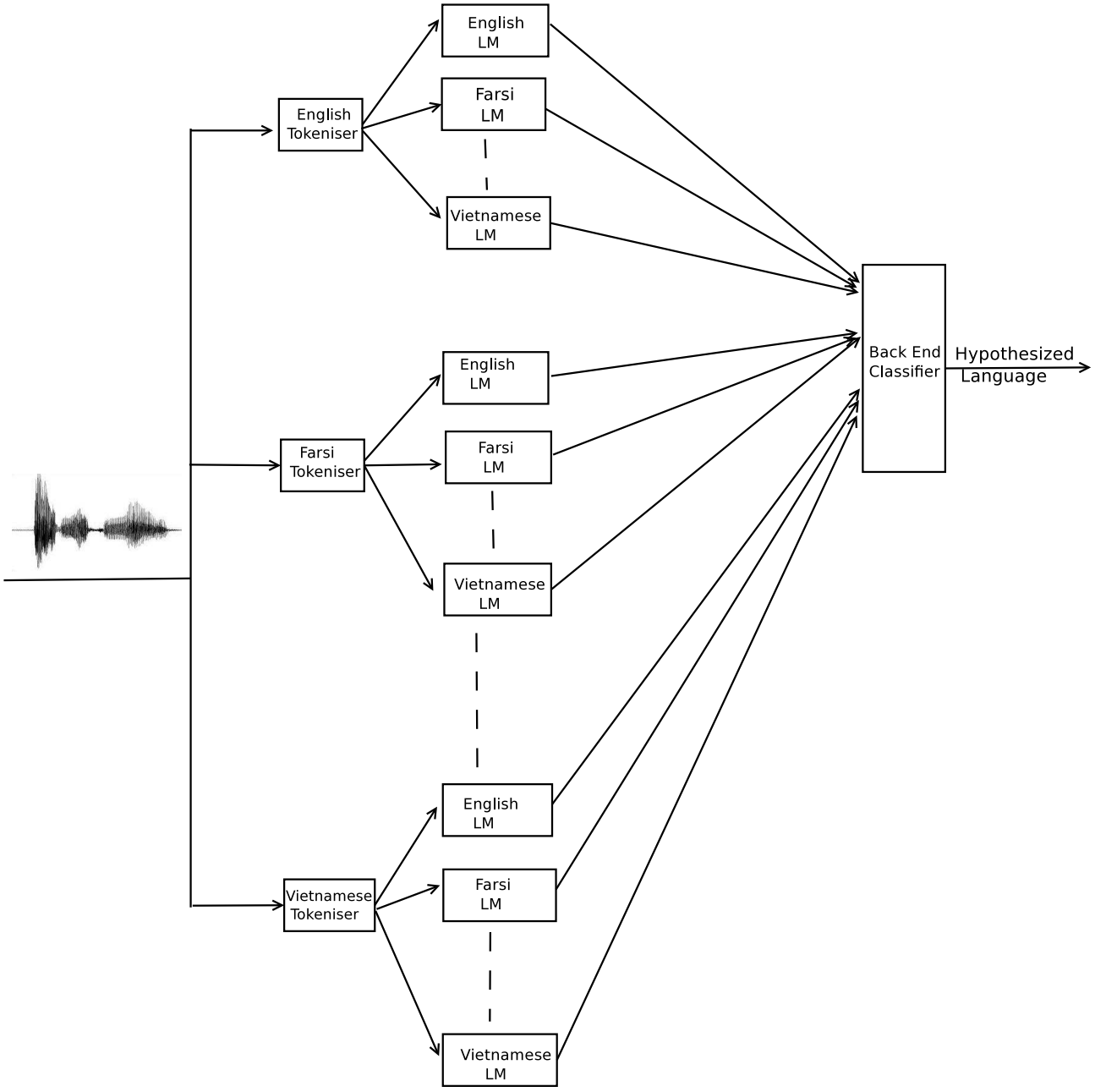

Fig. 4.2: Multiple Tokeniser used to decode speech utterance.

Table 4.1: *Performance of LID systems on OGI-MLTS*

LID Systems	Baseline System	Universal Syllable Model system
Histogram Approach	25.5 %	41.36 %
Histogram Approach, Training and Dev	26.81 %	45 %
Acoustic Scores	44.5 %	55 %
Acoustic + Histogram	48.18 %	57.72 %
Vector Space Approach	27.2 %	31.81 %

For histogram based approach, training data of each of the classes are used for building the language model. For computing the bigram statistics using interpolated language model, the bigram coefficient is set to 0.666, the unigram coefficient to 0.333 and offset to 0.001. The best performance is found to **25.5** %. To evaluate the performance of the system on the amount of data we have used training and development data to obtain the language model for each of the classes, it has been observed that the performance increased by just **1** %.

For the vector space based approach, we get 40 vectors containing bi-gram statistics for each of the classes after training. During testing, K-NN is used for classification. We have experimented with K = 1, 2, ..., 60 and observed that optimal performance

is obtained with K = 20 with accuracy of 27.2 %.

4.6.2 Universal Syllable Model

The class acoustic models are obtained by first building universal syllable models and then adapting the means of the models (multiple universal models). The parameters description is the same that is used in previous chapter and best performance being **55** % . The syllable language model is created by using each of the language data. Due to limited amount of data, we have used syllable uni-gram statistics.

For the histogram based approach, during testing, it is possible that the expected syllable may be in top-N list rather than top-1 list of syllables, therefore we consider top N scoring syllables. Each syllable is scored against the universal syllable models and N highest scoring syllable index is considered to build the histogram of the test utterance. We observed that the accuracy is **41.36** % with N = 2 and performance does not increase for N > 2. To evaluate the performance of the system on amount of data we take training and development data to extract the syllable uni-gram statistics. We observed that language identification accuracy increases significantly to **45** %.

For the vector space based approach, the best performance is **31.81** % for K = 20. It is observed that the accuracy of the system does not increase on increasing the training data.

4.6.3 Combining System

The acoustic system score (s_i^{acc}) and language model score is combined as given by equation:

$s_i = \alpha s_i^{acc} + (1 - \alpha) s_i^{lm}$

where, α is the weight-age parameter balancing the two scores and $0 < \alpha < 1$

We obtained LID accuracy of **48.18** % and **57.72** % for the baseline system and universal syllable based system respectively.

4.7 Conclusion

In this chapter, syllable as a token is used in building language identifications system. It was observed that histogram based approach performs better than vector space approach for syllable based LID system. The syllable based LID system performs better than GMM based system and as the training data increases the performance of syllable based system increases significantly. One of the advantages of GMM based system over syllable based LID system is that it is computationally efficient.

CHAPTER 5

IVector for Language Identification

5.1 Introduction

Most of the successful LID systems borrow ideas from speaker verification systems. The state of the art speaker verification systems uses total variability concepts to model the speaker. It closely resemble the general model, joint factor analysis technique used in the context of Gaussian Mixture Model.

In JFA, it is assumed that stacked means of an utterance is composed of the speaker supervector [1] and session super-vector and can be expressed by the following equation:

$$M_{utt} = m + Vy(s) + Ux_{utt} + Dz(s) \tag{5.1}$$

where

- M_{utt} is of size CF X 1 [2]
- m is the UBM super-vector.
- U, V and D are of size CF X K, CF X S, CF X CF and represent the session and speaker variability subspace.

However in I-vector based GMM speaker verification system, as found in [28] that the channel matrix contains speaker information. Hence the channel and inter-speaker

[1] A super vector is created by stacking the means of a GMM

[2] F is the feature size

and intersession variabilities are coupled together to form a single matrix, called total variability space. It can be expressed by the following equation

$$\boldsymbol{M_{utt}} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{w_{utt}} + \boldsymbol{\epsilon} \tag{5.2}$$

where $\boldsymbol{w_{utt}}$ is latent variable denoting a vector in the variability space and $\boldsymbol{T}$ is the basis vector of the reduced dimension variability space. It is assumed, $\boldsymbol{w_{utt}}$ follows Gaussian distribution with zero mean and unit variance i.e. $\boldsymbol{x_{utt}} \sim \mathrm{N}(\boldsymbol{0},\boldsymbol{I})$. The error term ϵ is the information that cannot be captured by the factor analysis model and assumed to have Gaussian distribution and diagonal covariance matrix. Hence, prior distribution of $\boldsymbol{M_{utt}}$ is given by

$$\boldsymbol{M_{utt}} \sim N(\boldsymbol{m}, \boldsymbol{T}\boldsymbol{T}') \tag{5.3}$$

i-vector has close resemblance with eigen voice as introduced in [29]. Both the approaches use EM algorithm to compute the parameters, the main difference is that in eigen-voice each of the speakers are described by one single latent variable whereas in i-vector each utterances represent one latent variable.

5.1.1 Parameter Estimation

The parameter estimation is similar to probabilistic principal component analysis as introduced in [30]. In such latent variable model, the expectation maximization problem consists of maximizing total data likelihood with respect to the posterior distribution of the latent variables as defined in equation 5.4

$$E_{X/\chi}([\sum_{n=1}^{n=N} p(\boldsymbol{h_n}, \boldsymbol{x_n})]) \tag{5.4}$$

Let us assume from 'H' number of feature vectors ($h_1,\ h_2,\ \cdots\ ,\ h_H$), N_c is the

frames aligned with the c^{th} mixture of the GMM and define the first order statistics as

$$F_c = \sum_{n=1}^{N_c} (h_n - m_c) \tag{5.5}$$

where m_c is the mean of the c^{th} GMM component. Algorithm to compute T matrix and i-vector of an utterance is given below

1. Randomly initialize T matrix and normalize with the ubm variance.
2. Estimate the i-vector of an utterance as

$$x_i = C_i^{-1} T' F_i \tag{5.6}$$

$$C_i = I + \sum_{c=1}^{K} N_x^c T^{c'} T^c \tag{5.7}$$

3. Compute the following statistics from the training data

$$C_a = \sum_i F_i x_i' \tag{5.8}$$

$$A^c = \sum_i (N_i^c (C_i^{-1} + x_i x_i') \tag{5.9}$$

4. Update the parameter i.e. T matrix as given by the following formula

$$T^c = C_a A^{c^{-1}} \tag{5.10}$$

5. Iterate steps 2-4 until convergence.

5.1.2 Scoring

For a test utterance, latent i-vector is extracted and cosine distance is used for scoring. However in language identification, a large number of utterances are available for

adaptation. hence average cosine distance is used as the score as defined by equation 5.11 and 5.12

$$cos(w_l, w_{utt}) = \frac{w_l^{'} w_{utt}}{|w_l||w_{utt}|} \tag{5.11}$$

$$d(l, utt) = \frac{1}{R} \sum_{i=1}^{i=R} cos(w_l^i, w_{utt}) \tag{5.12}$$

5.1.3 Linear Discriminant Analysis (LDA)and Within Class Covariance Matrix (WCCN)

LDA is a widely used technique to find dimension which maximally discriminates the classes. It consists of maximizing the inter-class variability while minimizing within class variability (or scatter). The problem ultimately reduces to finding eigen direction of the matrix $(\Sigma_b\ \Sigma_{wn})^{-1}$, where Σ_b is between class scatter matrix and Σ_{wn} is within class scatter.

WCCN was first introduced in [31] and used in the context of speaker verification and SVM. It consists of factorizing the W (equation 5.13) into BB'.

$$W = \frac{1}{C} \sum_{1}^{C} \frac{1}{n_i} \sum_{j} (w_j^i - \bar{w})(w_j^i - \bar{w})^{'} \tag{5.13}$$

5.1.4 Integrating syllable based system in ivector framework

A universal syllable models is build as explained in section **??**. The means of the universal models is stacked to obtain the ubm super-vector. The sufficient statistics are computed by scoring against this universal models by (i) segmenting the utterance into syllables, (ii) scoring each of the syllables against this universal models and (iii) align each of the syllables with the states of the highest scoring syllable model.

5.1.5 Experiments

i-vector experiments are performed in NIST 2005 LRE. The identification results are tabulated in table 5.1

5.1.5.1 ivector system

The best performance is obtained when the dimension of ivector = 400. Applying LDA on the ivectors of each of the languages, performance of the LID system increases by 25 %.

5.1.5.2 Syllable based system

The best performance of the top down clustering and universal syllable model is obtained with 1024 clusters. The integrated system (syllable based system in the ivector framework) performs worse than the universal syllable or top down clustering syllable model based LID systems

Table 5.1: *Performance of LID systems on NIST 2005 LRE*

LID Systems	Accuracy
i-vector	41.5 %
ivector + LDA	**66.34** %
Universal Syllable Model	45.76 %
Top Down Clustering	45.65 %
Integration of Syllable based system and ivector	41.67 %

5.2 Feature Switching Experiments

Feature selection is an important step in the development of any classification system. In this section, the diversity in four different feature representations (and their combinations) is exploited for language verification task. The principle behind the method is that using a single feature (or a combination of features) for all classes may not result in optimal classification, as a given feature may identify certain classes better than others. For certain classification problems, the classifier can utilise additional information to make the classification decision. For example, in a verification/authentication system, only the claimed class has to be processed for making the verification decision.

Study using four different features, namely: (a) the standard Mel-frequency cepstra (MFCC), (b) linear predictive cepstral coefficients (LPCC), (c) Fourier transform phase based modified group delay features (MODGDF) and (d) Mel-frequency slope (fSlope) is done.

For a given task, the method first determines which feature is best suited for each class. A framework in which the likelihood score for the class under consideration is computed using this well-suited feature. This results in using different features to score different classes, a paradigm is call ***feature-switching.*** Conventional systems typically have only one feature representation (or a combination thereof), which is used for all classes under consideration. Feature-switching is naturally extended to feature combinations as well. A mutual information based method, initially reported in [32] is presented to determine the most suitable feature for a class from training data.

5.2.1 Utilising information from multiple feature streams

Information from multiple feature streams can be combined in two ways:

- **Early fusion:** In this case, the combined representation is formed at the acoustic level by concatenating the different representations. The resultant high dimensional feature vector is used for training and evaluating the classifier. Only early fusion is considered in this paper.
- **Late fusion:** In this case, the combined representation is formed at the decision or score level. This requires building two or more individual classification systems using the features under consideration and combining their decisions or scores.

Combination of information from multiple feature representations generally improves the performance of the classifier. Feature-switching aims to obtain performance comparable to feature combination without its overhead.

5.2.2 Exploiting feature diversity

We call a feature which is effective at identifying a particular class an *optimal-feature* for that class. From the post-mortem analysis for each task, we determine that different classes have different optimal-features. It is seen that there is least error when optimal-features are used for the identification. To improve accuracy in any identification task, we can attempt to use feature-switching as follows. We can compute distortion values or scores always using optimal-features for each of the candidate classes under consideration. We then classify the test sample as the class with the higest score.

5.2.3 Issues in implementing feature-switching

One practical limitation implementing feature-switching as above is that the numerical range of scores or distortion values are not directly comparable across features. Thus we cannot apply feature-switching directly for emotion, language or speaker identification. A possible solution to this could be some form of score normalization to bring the scores into comparable numeric range. This is still under investigation.

5.2.3.1 Application to verification

Since direct comparison of scores is not possible across feature streams, we can apply feature-switching in a verification scenario. In this case, we need to compute the score only for the claimed class, and hence avoids the need for comparison with other scores. Feature-switching will result in using different features for different claims. Thus, in the remaining sections of this paper, the effectiveness of feature-switching is shown for verification tasks.

The objective of feature-switching is to make use of the information in different feature spaces. When compared to feature-level combination methods, the proposed method is computationally more effective, since we neither need to extract two features for every test case, nor a need to work in high dimensional feature spaces.

5.2.4 Determining the optimal-feature for a class

Thus, feature-switching involves in determining the optimal-feature for each class. Note that 'class' means the emotion classes in a emotion verification task, language classes in a language verification task, and speakers in a speaker verification task. To avoid using a post-mortem analysis to determine the optimal-feature for each class, we need to use train/development data. We present below an information theoretic

method to determine the optimal-feature for a given class.

For classification tasks, one must consider two aspects of feature representation: (a) the ability to capture maximum information from the acoustic space into the feature space (representative property) and, (b) the ability to discriminate between different classes (discriminative property).

5.2.5 Verification framework

Figure 5.1 shows the architecture of the proposed verification framework. Note that this could be a speaker or language or emotion verification system.

In the training phase, the optimal-feature is determined for each class using the optimal feature function (**??**). The ⟨class, optimal-feature⟩ pair is stored in a lookup table (LUT), which is indexed by class identity. For example, in a speaker verification system, the LUT consists of ⟨speaker, optimal-feature⟩ entries, whereas for language verification, the LUT will contain ⟨language, optimal-feature⟩ entries. The LUT contains an entry for each class in the system. Different parameters of (**??**) result in different LUTs for the same set of classes.

In the evaluation phase, the optimal-feature of the claimed class is determined from the lookup table. The optimal-features are extracted from the input speech waveform. The likelihood score is computed against the corresponding model and the verification decision is made. This results in the verification system performing feature-switching, by extracting different features for different claims.

5.2.6 Experiments

The method of feature-switching is applied to speaker, language and emotion verification. In the databases mentioned in section **??**, the NIST 2003 SRE is already

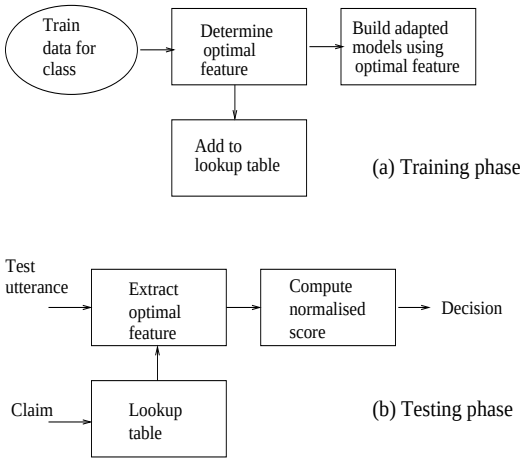

Fig. 5.1: The proposed verification system incorporating feature switching. (a) Training phase and (b) testing phase.

a verification dataset. In the experiments descibed below, the emotion dataset and language datasets are used in verification mode. Each test utterance in the respective databases is verified against all classes in that database.

5.2.6.1 Baseline results

The equal error rates (EER) for each task using different features and feature combinations is given in table 5.2.

It is seen that the best baseline performance for language verification is XXX, for emotion it is XXX and speaker verification it is XXXXX.

5.2.6.2 Implementing feature-switching

In the experiments below, the feature-switched speaker verification system was implemented using the optimal-feature function in equation **??**. The emotion and language verification systems were built with post-mortem analysis on test data from the respective identification systems.

System	EER (%)
Language verification system	
MFCC	45.90
MODGDF	42.72
fSlope	35.45
LPCC	45.00
MFCC+MODGDF	45.00
MFCC+fSlope	42.27
MFCC+LPCC	46.36
MODGDF+fSlope	41.36

Table 5.2: Verification EERs for the three tasks. Early fusion results of some feature combinations are also given.

For optimal performance, the weight given to the representative and discriminative measures in Equation **??**, is dependent on the class. Thus the weight α are different for different classes. This suggests that, for each class, the amount of representative and discriminative information for good verification performance is different. Thus, these weights have to be determined empirically.

The idea of feature-switching can be naturally extended to feature combinations by feature-switching between them. Feature-switching can also be done between individual features and early fusion of features (for example, in this case, some claims are verified using a single feature like MFCC whereas other claims by early fusion of fSlope and MFCC.)

The EERs for the three systems using feature-switching is given in Table 5.3. It is seen that feature-switching does indeed improve performance when compared to the

baseline systems.

System	EER (%)
Language verification system	
MFCC/MODGDF	28.64
MFCC/LPCC/MODGDF	28.01
MODGDF+fSlope/MFCC+fSlope	26.37
MFCC+LPCC/MODGDF/fSlope	25.46

Table 5.3: Equal error rates for the three tasks using feature-switching. In some cases, feature-switching is done between early fusion of features.

5.3 Conclusion

CHAPTER 6

Conclusion

BIBLIOGRAPHY

[1] Y. K. Muthusamy, K. M. Berkling, T. Arai, R. A. Cole, and E. Barnard, "A comparison of approaches to automatic language identification using telephone speech," in *EUROSPEECH*, 1993.

[2] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech.," in *IEEE Transaction Speech and Audio Proceeding.*, pp. 31–44, 1996.

[3] P. Matejka, L. Burget, O. Glembek, P. Schwarz, V. Hubeika, M. Fapso, T. Mikolov, O. Plchot, and J. Cernocký, "But language recognition system for nist 2007 evaluations," in *INTERSPEECH*, pp. 739–742, 2008.

[4] R. Tong, B. Ma, H. Li, and E. Chng, "Selecting phonotactic features for language recognition," in *INTERSPEECH*, pp. 737–740, 2010.

[5] X. Zhou, J. Navratil, J. W. Pelecanos, G. N. Ramaswamy, and T. S. Huang, "Intersession variability compensation for language detection," in *ICASSP*, pp. 4157–4160, 2008.

[6] Nagarajan, T., *Implicit Systems for Spoken Language Identification*. PhD dissertation, Indian Institute of Technology Madras, Department of Computer Science and Engg., Madras, India, May 2004.

[7] T. J. Hazen and V. W. Zue, "Automatic language identification using a segment-based approach," in *EUROSPEECH*, 1993.

[8] P. Matjka, *Phonotactic and acoustic language recognition*. PhD thesis, 2009.

[9] B. M. Haizhou Li and C.-H. Lee, "A vector space modeling approach to spoken language identification," *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 15, no. 1, pp. 271–284, 2007.

[10] P. A. Torres-carrasquillo, E. Singer, M. A. Kohler, and J. R. Deller, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Proc. ICSLP*, pp. 89–92, 2002.

[11] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[12] J. Navratil and D. Klusacek, "On linear dets," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, pp. IV–229 –IV–232, april 2007.

[13] Y. K. Muthusamy and R. A. Cole, "Automatic segmentation and identification of ten languages using telephone speech," in *ICSLP*, 1992.

[14] Y. K. Muthusamy and R. A. Cole, "A segment-based automatic language identification system," in *NIPS*, pp. 241–248, 1991.

[15] I. R. T. Schultz and A. Waibe, "Lvcsr-based language identification.," in *IEEE International Conference on In Acoustics, Speech, and Signal Processing.*, p. 781784, 1996.

[16] P. A. Torres-carrasquillo, D. A. Reynolds, J. Deller, and Jr., "Language identification using gaussian mixture model tokenization," *IEEE ICASSP*, vol. 2002, pp. 757–760.

[17] A. K. V. S. Jayram, V. Ramasubramanian, and T. V. Sreenivas, "Automatic language identification using acoustic sub-word units," in *INTERSPEECH*, 2002.

[18] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 210–229, 2006.

[19] R. Crdoba, R. San-segundo, J. Macas, J. M. Montero, R. Barra, L. F. Dharo, and J. C. Plaza, "Integration of acoustic information and pprlm scores in a multiple-gaussian classifier for language identification," *IEEE Odyssey*, 2006.

[20] R. de Crdoba, L. F. D'Haro, F. Fernndez, J. Montero, and R. Barra, "Language Identification using several sources of information with a multiple-Gaussian classifier," in *Proc. of Interspeech*, pp. 2137–2140, September 2007.

[21] W. M. Campbell, "A covariance kernel for svm language recognition," in *ICASSP*, pp. 4141–4144, 2008.

[22] "Kl divergence." `http://en.wikipedia.org/wiki/Kullback-Leibler_divergence`, 2003.

[23] T. Nagarajan and H. Murthy, "Language identification using parallel syllable-like unit recognition," in *ICASSP*, pp. 401–404, 2004.

[24] V. K. Prasad, T. Nagarajan, and H. A. Murthy, "Automatic segmentation of continuous speech using minimum phase group delay functions," *Speech Communication*, vol. 42, no. 3-4, pp. 429–446, 2004.

[25] "Hierarchical clustering." `http://en.wikipedia.org/wiki/Hierarchical_clustering`, 2003.

[26] Y. K. Muthusamy, R. A. Cole, and B. T. Oshika, "The ogi multi-language telephone speech corpus," pp. 895–898, 1992.

[27] "2003 nist lre plan." `http://www.nist.gov/speech/tests,2003`, 2003.

[28] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[29] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, 2005.

[30] C. M. Bishop, "Bayesian pca," in *NIPS*, pp. 382–388, 1998.

[31] A. O. Hatch, S. S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition," in *INTERSPEECH*, 2006.

[32] R. Padmanabhan and H. A. Murthy, "Acoustic feature diversity and speaker verification," in *INTERSPEECH*, pp. 2110–2113, 2010.

LIST OF PUBLICATIONS

Publications

- Subhadeep Dey, Rajeev Rajan, Padmanabhan R. and Hema Murthy, "Feature Diversity for Emotion, Language and Speaker Verification", NCC 2011, IISc Bangalore.
- Subhadeep Dey and Hema Murthy, "Universal Syllable Tokeniser for Language Identification", NCC 2012.
- Subhadeep Dey and Hema Murthy, "Unsupervised Clustering of syllables for language identification", EUSIPCO 2012.