AUTOMATIC SPEAKER AND LANGUAGE IDENTIFICATION

A First Year Report
Submitted to the School of Computer Engineering
of the Nanyang Technological University

by

Tong Rong

for the Confirmation for Admission
to the Degree of Doctor of Philosophy

March 28, 2006
Abstract

This thesis deals with the problem of automatic language identification (LID) and automatic speaker identification (SID) given the speech signal as input. Both researches have received renewed interest due to heightened homeland security awareness, e.g. in the use of speaker’s voice print for biometric identification, language identification for the classification of speech archives and call-scanning. In addition, technological progress on speaker and language identification (SLID) research will also enhance the robustness of automatic speech recognition.

The technologies used in SLID applications are closely related as both apply statistical models to identify speaker and the language identity. The key to solve the SLID problem is the detection and exploitation of differences between languages and speakers. The speaker and languages differ from one another along many dimensions, these include, phoneme inventory, phonotactic, prosodics, syllable structure, lexical words and grammar. These differences must be identified and extracted from raw speech data to successfully attack the SLID issues.

While the progress of speaker and language identification research in the last few years has been heartening, significant improvements is still required for real-world environment applications as available data from these environments is often noisy and/or has short duration. Although human can normally successfully apply SLID using these data, current systems are unable to perform as robustly. This indicates that current systems have not fully exploited all available information in the recording and improvements can be made.

The first year of the research focused on general literature study and investigating different approaches in language identification techniques. The research is focused on investigating the discriminative features and their fusion in language identification. We
examine the fusion of five features at different levels of abstraction for language identification, including spectrum, duration, pitch, n-gram phonotactic, and bag-of-sounds features.

Our experimental results show that different levels of information provide complementary language cues. The prosodic features are more effective for shorter utterances while the phonotactic features work better for longer utterances. For the task of twelve languages, our current system with fusion of five features achieved 2.38% Equal Error Rate (EER) for 30-seconds speech segments on NIST 1996 dataset. Our system has also participated in NIST 2005 Language Recognition Evaluation (LRE). The 2005 NIST Language Recognition evaluation (LRE-05) is part of an ongoing series of international benchmarking of language technologies organized by National Institute of Standards and Technology (US) and sponsored by National Security Agency (US). LRE-05 was participated by 15 teams from 11 countries and our language recognition system won 4 titles, specifically:

- First place in Korean Language Recognition
- Second place in English Language Recognition
- Third place in All Language Trials
- Third place in Primary Language Condition
Acknowledgments

I would like to thank my supervisors, Dr. Li Hai Zhou (I2R) and Dr. Chng Eng Siong (NTU). I am grateful to them for introducing me to the area of speech processing, and helping me in my research. As a part time student, I often struggle between the study and work, Dr. Li and Dr. Chng always support me and give me lots of freedom on my schedule.

I would also like to thank my college, Dr. Ma Bin, for the helpful discussion and valuable advice. I’m also very grateful to my college Dr. Zhu Dong Lai, who has always give me support and help, especially in the 2005 NIST language identification competition. Thanks to all my colleges in Speech and Dialogue Processing lab, they always tolerant me when I used up all the computation machines. Special thanks to Mr. Xiao Xiong, who has help me in the graduate school related matters.

Finally, I would like to thank my family, I have always felt their presence by my side whenever I meet problems. My work would not have been possible without their affection and constant encouragement.
# Contents

Abstract ......................................................... i  
Acknowledgments ................................................. iii  
List of Figures .................................................... vi  
List of Tables ..................................................... vii  
List of Abbreviations ............................................. viii  

1 Introduction ........................................ 1  
   1.1 Research Objectives .................................... 1  
   1.2 Language Identification ................................ 2  
   1.3 Contributions .......................................... 2  
   1.4 Organization of the Report ............................. 3  

2 Overview of Language Identification ............... 5  
   2.1 Background ............................................. 5  
   2.2 Distinguishing Between Languages .................... 6  
   2.3 Probabilistic formulation of language identification problem ................ 8  
   2.4 The Process of Languages Identification .......... 9  
   2.5 Performance Measurement .............................. 10  

3 Related Works .............................................. 13  
   3.1 Overview ............................................... 13  
       3.1.1 1996 and 2003 NIST LRE ......................... 14  
   3.2 Feature Extraction .................................... 14  
       3.2.1 Shifted Delta Cepstra(SDC) ...................... 15  
       3.2.2 Long-time TempoRAI Patterns(TRAP) ......... 16
3.3 Five state-of-the-art LID systems .................................. 18
   3.3.1 Phonotactic based LID system ............................... 19
   3.3.2 Cepstral based LID system ................................. 26

4 Proposed LID System ............................................. 32
   4.1 Overview ..................................................... 32
   4.2 Language Identification Corpora .............................. 33
   4.3 LID on Acoustic, Prosodic and Phonotactic Features ....... 34
       4.3.1 N-gram LM in P-PRLM .................................. 34
       4.3.2 Bag-of-Sounds (BOS) .................................... 36
       4.3.3 SDC Feature in GMM .................................... 38
       4.3.4 Duration .................................................. 39
       4.3.5 Pitch ...................................................... 41
   4.4 Experiment Results ............................................ 43
       4.4.1 Individual System Performance .......................... 43
       4.4.2 Analysis on P-PRLM and BOS approach ................. 44
       4.4.3 System Fusion ............................................. 45
       4.4.4 Final Decision Classifiers .............................. 47

5 Conclusions and Future Work ..................................... 49
   5.1 Conclusions ................................................... 49
   5.2 Future Works ................................................ 50
   5.3 Schedule of Future Research .................................. 50

Publication ......................................................... 53

References ......................................................... 54
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>LID process</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>Shifted delta cepstral coefficients</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>Conventional feature extraction</td>
<td>17</td>
</tr>
<tr>
<td>3.3</td>
<td>TRAP feature</td>
<td>18</td>
</tr>
<tr>
<td>3.4</td>
<td>PRLM diagram</td>
<td>20</td>
</tr>
<tr>
<td>3.5</td>
<td>PPRLM system</td>
<td>22</td>
</tr>
<tr>
<td>3.6</td>
<td>BOS system</td>
<td>23</td>
</tr>
<tr>
<td>4.1</td>
<td>Five levels of LID features</td>
<td>32</td>
</tr>
<tr>
<td>4.2</td>
<td>Five LID systems</td>
<td>34</td>
</tr>
<tr>
<td>4.3</td>
<td>DET curve of fused system on NIST 2003 LRE (3-sec)</td>
<td>47</td>
</tr>
<tr>
<td>5.1</td>
<td>Schedule for future research</td>
<td>51</td>
</tr>
</tbody>
</table>
List of Tables

3.1 EER of NIST 1996 LRE .................................................. 19
3.2 EER of NIST 2003 LRE .................................................. 19

4.1 EER of Individual system in 1996 LRE ................................. 43
4.2 EER of Individual system in 2003 LRE ................................. 44
4.3 EER of front-end and back-end combination in 1996 LRE .......... 44
4.4 EER of front-end and back-end combination in 2003 LRE .......... 45
4.5 Speed of frontend and backend combination in 1996 LRE 30 seconds . 45
4.6 EER of system fusion on 1996 LRE ...................................... 46
4.7 EER of system fusion on 2003 LRE ...................................... 47
4.8 EER reduction by member classifiers in the ensemble ............... 48
4.9 P-PRLM 30s EER on the number of mixtures for true model versus negative model .............................................. 48
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID</td>
<td>Language Identification</td>
</tr>
<tr>
<td>SID</td>
<td>Speaker Identification</td>
</tr>
<tr>
<td>LM</td>
<td>Language Model</td>
</tr>
<tr>
<td>EER</td>
<td>Equal Error Rate</td>
</tr>
<tr>
<td>FAR</td>
<td>False Acceptance Rate</td>
</tr>
<tr>
<td>FRR</td>
<td>False Rejection Rate</td>
</tr>
<tr>
<td>DET</td>
<td>Detection Error Tradeoff</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>OGI-TS</td>
<td>Oregon Graduate Institute Multi-Language Telephone Speech Corpus</td>
</tr>
<tr>
<td>IDR</td>
<td>Identification Rate</td>
</tr>
<tr>
<td>LRE</td>
<td>Language Recognition Evaluation</td>
</tr>
<tr>
<td>GMM</td>
<td>Gaussian Mixture Model</td>
</tr>
<tr>
<td>PRLM</td>
<td>Phone Recognition followed by Language Modeling</td>
</tr>
<tr>
<td>P-PRLM</td>
<td>Parallel Phone Recognition followed by Language Modeling</td>
</tr>
<tr>
<td>LMSI</td>
<td>Laboratoire d’Informatique pour la Mecanique et les Sciences de l’Ingenieur</td>
</tr>
<tr>
<td>LPC</td>
<td>Linear Predictive Coefficients</td>
</tr>
<tr>
<td>MFCC</td>
<td>Mel-Filterbank Cepstral Coefficient</td>
</tr>
<tr>
<td>SDC</td>
<td>Shifted Delta Cepstra</td>
</tr>
<tr>
<td>TRAP</td>
<td>TempoRAi Patterns</td>
</tr>
<tr>
<td>BUT</td>
<td>Brno University of Technology</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>BOS</td>
<td>Bag of Sounds</td>
</tr>
<tr>
<td>GLDS</td>
<td>Generalized Linear Discriminant Sequence</td>
</tr>
<tr>
<td>UBM</td>
<td>universal background model</td>
</tr>
<tr>
<td>ASR</td>
<td>Automatic Speech Recognition</td>
</tr>
<tr>
<td>EM</td>
<td>Expectation Maximization</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Research Objectives

This research concerns the use of speech as input for automatic language identification (LID) and automatic speaker identification (SID). Both applications have received renewed interest due to heightened homeland security awareness, e.g. Speaker’s voice print for biometric identification [1] [2] [3] and Language identification for spoken language translation [4] [5], multilingual speech recognition [6] [7] and spoken document retrieval [8]. In addition, progress on speaker and language identification research will also enhance the robustness of automatic speech recognition.

The technologies used in speaker identification and language identification applications are closely related as both apply statistical models to identify speakers [9] [10] [11] and language[12] [13] [14] from recorded speech data. Some fundamental speech processing techniques are used in both speaker and language identification tasks. For example, speech segmentation [15], speech representation [16], Feature extraction and selection [17], speech tokenization [18]. Nevertheless many of the approaches to spoken language identification have adopted techniques used in current speaker-independent speech recognition systems. For example, one of the most popular method in speaker recognition is to use Gaussian mixed model to model the cepstral features [48] [49] [11]. This method is also widely applied in the language identification task [30] [19].

The first year of my study had focus on exploring the methods in language identification. We believe that the findings in the language identification will also benefit the study in speaker identification.
1.2 Language Identification

The fundamental issue of the automatic language identification is to explore the effective discriminative cues for languages. Recent studies have explored different levels of speech features which include articulatory parameters [21], spectral information [22], prosody [23], phonotactic [22] and lexical knowledge [24]. It is generally believed that spectral feature and phonotactic feature provide complementary language cues to each other [28]. Human perception experiments also suggest that prosodic features are informative language cues [21]. Within the first few days of life, infants are able to differentiate between the sounds of their native language and other languages, possibly by attending to prosodic properties of speech [66]. These phenomena indicate that the listener in the absence of higher level knowledge of a language, he presumably relies on lower level constraints. However, prosodic feature has not been fully exploited in LID [29].

In general, LID features fall into five groups according to their level of knowledge abstraction: syntactic, lexical, phonotactic, prosodic and acoustic. Lower level features, such as acoustic spectral feature, are easy to obtain but not robust as speech variations due to speaker or channel variations are present. Higher level features, such as lexical/syntactic features, rely on large vocabulary speech recognizer, which is language and domain dependant, and is therefore difficult to generalize across languages and domains. Phonotactic features which refers to the rules that govern the combinations of the different phones in a language, become a trade-off between computational complexity and performance. It is generally agreed that phonotactics, i.e. the rules governing the sequences of admissible phone/phonemes, carry more language discriminative information than the phonemes themselves. They are extracted from output of a phoneme recognizer, which is supposed to be more robust against effects such as speaker and channel than spectral features. Current researchers have focused on acoustic-prosodic-phonotactic features [28] [29] [67].

1.3 Contributions

The first year of this research had focused on investigating the discriminative features for language identifications. We study the fusion of five features at different level of
abstraction for language identification. These features are the spectrum, duration, pitch, n-gram phonotactic, and bag-of-sounds features. In addition, we also examine how the different levels of language cues complement the LID task.

We typically represent a speech utterance as a collection of independent spectral feature vectors. The collection of vectors can be modeled by a Gaussian mixture model, known as GMM [30] that captures the spectral characteristics of a language. The prosody of speech can be characterized mainly by energy, pitch and duration among others. They can be modelled in a similar way as that for spectral feature. Phonotactic features capture the lexical constraint of admissible phonetic combination in a language, e.g. one typical implementation is the P-PRLM (Parallel Phone Recognition followed by Language Model) approach that employs multiple phoneme recognizers which will tokenize a speech waveform into phoneme sequences and then characterizes a language by a group of n-gram language models (LM) over the phoneme sequence [22]. A new phonotactic model, known as bag-of-sounds was proposed recently to model utterance level phonotactics collectively and this technique's language discriminative performance is comparable to that of the n-gram LM [31], [32].

Our experimental results show that different levels of information provide complementary language cues. The prosodic features are more effective for shorter utterances while the phonotactic features work better for longer utterances. For the task of twelve languages identification, our system with fusion of five features achieved 2.38% Equal Error Rate (EER) for 30-seconds speech segments on NIST 1996 dataset. Our system also participated in NIST 2005 LRE and had achieved good results.

1.4 Organization of the Report

The rest of the report is organized as follows:

Chapter 2 presents an overview of the language identification task. Firstly we briefly review the applications that require the language identification, then we analyze the discriminative cues that human being are using to distinguish languages, after that we introduce the probabilistic formulation of language identification and the general process of a language identification system. The performance measurement methods for LID is described in the latter section.
Chapter 3 reviews the related works in automatic language identification. We discuss two feature extraction approaches for LID and introduce six state-of-the-art LID systems.

Chapter 4 presents our proposed method in language identification. LID systems that based on five features are introduced, experiments results are presented to compare the accuracy of the individual method. The performance of the fused system are also reported.

Finally, chapter 5 gives the conclusion of the report and presents the directions for further work of the research.
Chapter 2

Overview of Language Identification

2.1 Background

Automatic language identification is an important issue in the global community. As collaborations and trading between different countries are more and more popular, people that speak different languages must be able to communicate seamlessly, to communicate with each other is necessary to know language with which we are dealing, a LID system can help to lower the communication barriers among people from different regions. The tourist agency may need automatic language identification to quickly find out what language that the customer prefers and provide more efficient service; Due to the terrorist attack in the recent years, the homeland security become high importance in all countries, the automatic language identification technology can be used to pre-process and filter the suspected speech. In a multilingual country like Singapore, there are four major languages used in the day-to-day life, the automatic LID systems have special significance. For example, a hospital automatic appointment booking system will need LID technology to automatic detect what language the caller is using, therefore it can automatic switch to the language that caller prefers, this will greatly improve the quality of the booking system.

Numerous applications for LID fall in two main categories: pre-processing for machines and pre-processing for human listeners [22]. One example of the first category is a multilingual voice controlled information retrieval system. An example of the second category is the language identification system used to route an incoming telephone call to a human operator fluent in the corresponding language.
Although the research in LID has made great progress in recent years, it is still a challenge to build a reliable practical LID system. In the following sections, we first discuss the possible information sources that can be utilized in language identification, then we present the theoretical formulation and general process of the LID system, finally the performance measurement method of LID system is discussed.

2.2 Distinguishing Between Languages

As with speech recognition, humans are the most accurate language identification system in the world today [21]. Within a second of hearing a conversation, people are able to determine if it is language we know. Even if the language is unknown to us, we can often make subjective judgement about the language, e.g. “it sounds like Japanese”. To solve the language identification problem, we could start by looking at the features that human use to distinguish languages.

There are a variety of cues that humans use to distinguish between languages [5] [22] [35], they include:

**Acoustic.** Acoustic is the physical characteristics of the speech signal, it is defined by frequency, time and intensity of the speech. Typically, the acoustic information of a spoken utterance is represented as a sequence of feature vectors where each individual vector represents the acoustic information for a particular time frame. As these feature vectors may contain ‘noisy’ information for language identification, e.g., channel, speaker related noise, how to extract robust acoustic features effectively is a challenge to acoustic feature based LID system.

**Prosodics.** Prosody refers to acoustic structure that extends over several segments. The stress, intonation (pitch contour), and rhythm (the duration of phones, speech rate) are all important elements used within the prosodic structure of a spoken utterance. The manner in which these elements are incorporated into the prosodic structure of an utterance varies across languages. The differences across languages can often be observed in the realization of the prosodic features which determine the tones or stress contained throughout an utterance. For example, tonal languages such as Mandarin have very different intonation characteristics compared with the stress languages such as English.
Phonotactics. There are various phonological factors which help define the distinctiveness of a language. Some of these factors include the phone set and the phonotactic constraints. Phonotactics refers to the rules that govern the combinations of the different phones in a language. There is a wide variance in phonotactics rules across languages. Different languages may have different rules governing how sequences of phonemes may be constructed. These phontactic constraints may result in having certain phonetic sequences similar to some languages but very different to others. For example, Japanese has strict phonotactic constraints which generally prohibit consonants from following consonants. English, on the other hand, has looser constraints which allow for the possibility of multiple consonants in succession. Hence phonotactics information can be used to capture some of the dynamical nature of speech lost during feature extraction. Nowadays, many successful LID systems often take advantages of phonotactics information.

Vocabulary. Conceptually, the most important difference between languages is that they use different sets of words, that is their vocabularies differ. Thus, a non-native speaker of English is likely speaking English with his/her native language prosodic patterns and phonemic inventory will still be judged to speak English by the vocabulary used.

Grammar. The ways in which words can be legally strung together also posses distinctive information. Even when two languages share a word, e.g. the word “bin” in English and German, the set of words which can precede and follow the word “bin” is different.

The differences in the above features must be identified and extracted from raw speech data to successfully attack the LID issues. While higher level information, such as words and grammar, may be used to uniquely determine the language of an utterance, utilization of these information for a large set of potential languages could be computationally expensive and may not realizable for real implementation. In this thesis, we will focus on the examination of lower level features, such as the acoustic, prosodic and phonotactics for accurate language identification features.
2.3 Probabilistic formulation of language identification problem

Let \( S = \{s_1, s_2, s_3, \ldots, s_n\} \) represent a sequence of acoustic segments corresponding to any of the language in the set of \( M \) languages \( L = \{L_1, L_2, L_3, \ldots, L_M\} \). The task of the LID is to determine the most likely language \( L^* \) of the input speech, specifically:

\[
L^* = \arg \max_i P(L_i|S) \tag{2.1}
\]

where \( P(L_i|S) \) is the posterior probability of language \( L_i \). Assuming that the input vector \( S \) belongs to one of \( M \) classes \( L_i, 1 \leq i \leq M \), the LID problem is a standard pattern classification problem. The main objective of this pattern classification is to decide which class the given vector \( S \) belongs to.

According to Bayes rule [69]:

\[
P(L_i|S) = \frac{P(S|L_i)P(L_i)}{P(S)} \tag{2.2}
\]

As remove \( P(S) \) does not affect the classification decision, the problem can be reduced to maximizing the joint density \( P(S|L_i)P(L_i) \). According to the rule given in (2.1), the objective is to choose the class \( L^* \) for which the posterior probability \( P(L_i|S) \) is maximum for a given \( S \). The can also be implemented using

\[
L^* = \arg \max_i P(S|L_i)P(L_i) \tag{2.3}
\]

where \( P(S|L_i) \) represents the likelihood probability of \( S \) corresponding to language \( L_i \) and \( P(L_i) \) denotes the a priori language probability, which is often assumed to be uniform for all languages, and hence can be ignored. Therefore the problem is simplified to

\[
L^* = \arg \max_i P(S|L_i) \tag{2.4}
\]

Thus the LID problem becomes the estimation of the posterior probability as in Eq.(2.1) or the likelihood probability as in Eq.(2.4).
Chapter 2. Overview of Language Identification

2.4 The Process of Languages Identification

As discuss in the last section, language identification tasks mainly involve three stages namely, feature extraction, training and testing as illustrated in Figure 2.1. The performance of the language identification systems are influenced by all the three stages.

The feature extraction module extracts the language discriminative information and filter those information that not related to language identification. During training, speech messages from one or more languages are analyzed, resulting in one or more models for each language. During testing, a previously unseen test message is applied to the system, and the system outputs the language associated with the model that most closely matches the test message [27].

![Figure 2.1: LID process](image)

The feature extraction module converts the continuous speech signal into a string of acoustic segments $S$. Most algorithms used for feature extraction in LID are adopted from speech recognition. The most common features are Linear Prediction Coefficient (LPC) [26] and Mel-Filterbank Cepstral Coefficient (MFCC) [25]. A relatively new feature is the TempoRAi Patterns (TRAP) feature that has also been successfully used in LID [40]. Another novel feature extraction methods: Shift Delta Cepstra (SDC) is proposed recently specially for language identification [30]. The details of TRAP and SDC feature will be discussed in the next Chapter.
Chapter 2. Overview of Language Identification

The process of building models from the features is termed as training. The training process varies according to system design. The simplest form of training uses sampled speech feature vectors, and the true identity of the language being spoken. Sampled speech segments from certain target languages are used to train a model that represents the target language. Such systems are analogous to pre-defining a set of templates and choosing the template that most closely matches the test pattern. The conventional Gaussian mixture based system belongs to this category. More complex approaches convert the feature vectors into some intermediate symbol representation, for example, the Phone Recognition followed by Language modelling (PRLM) system converts the raw feature vectors into phoneme sequences first and then prepare the language model training. Such methods are obviously computationally more costly and time consuming, but they do achieved better accuracy [22].

In the testing stage, the models are tested with the features from the test utterances for identifying the language. Different types of testing exist. The simplest testing process is straightforward: the unknown speech is evaluated by each model trained in the training process, the output scores are compared to make judgement. More complex approaches do not make decisions on the output score directly, they instead apply the pattern classification methods on the output scores from all modules. Here we call it the final classifier.

2.5 Performance Measurement

The general area of language identification consists of two fundamental tasks: Identification and Verification. Identification is the process of determining which language it is using provides a given speech signal. The system performs a 1:N classification. Generally it is assumed that all the signals are come from a fixed set of known languages, thus the task is often referred to as closed-set identification. Verification, on the other hand, is the process of accepting or rejecting that the given speech signal is in certain language. Since it is generally assumed that not all the languages are known to the system, this is an open-set task. By adding a “none-of-the-above” option to closed-set identification task one can merge the two tasks for what is called open-set identification.
The performances of the LID methods are evaluated as either the equal error rate (EER) for language verification or identification rate (IDR) for language identification. In the NIST LRE competition, the performance is reported as the value of Detection Error Tradeoff (DET) curve [20]. We will describe these measurements in the following paragraphs.

EER is used to decide the operating point where the false acceptance rate (FAR) and false rejection rate (FRR) are equal. It is a measurement for language verification. FAR is the percentage of the test utterances that are not in certain languages but are classified as 'true' for those languages. It also called false alarm rate. FRR is the percentage of the test utterances that in certain languages but classified as 'false' for those languages. Sometimes we also call it as miss classification rate. A LID system predetermines the threshold values for its false acceptance rate and its false rejection rate, and when the rates are equal, the common value is referred to as the equal error rate. The value indicates that the proportion of false acceptances is equal to the proportion of false rejections. The lower the equal error rate value, the higher the accuracy of the LID system.

For a given language $L$, the IDR is defined as:

$$IDR = \frac{n}{N}$$  \hspace{1cm} (2.5)

where $n$ is the number of correctly identified utterances in language $L$. $N$ is the total number of utterances in language $L$. A high IDR value represents high accuracy of the LID system.

The DET curve defines the minimum cost point as follows:

$$C_{DET} = (C_{Miss} \times P_{Miss|Target} \times P_{Target}) + (C_{FalseAlarm} \times P_{FalseAlarm|Non-Target} \times P_{Non-Target})$$  \hspace{1cm} (2.6)

where $C_{Miss}$ and $C_{FalseAlarm}$ represent the relative costs of a miss and a false alarm, respectively. $P_{Miss|Target}$ is the probability of miss for all target tests, while $P_{FalseAlarm|Non-Target}$ is the probability of false alarm for all the none target tests. $P_{Target}$ is the a priori probability of the target and $P_{Non-Target}$ is $1 - P_{Target}$, which refers to the probability of the non-target.

The cost function $C_{DET}$ will be evaluated for every point along the DET curve, the goal is to find the point where the function takes the minimum value. This point defines
the threshold for which the accuracy of the algorithm is optimum. Small value of DET shows good LID performance.

The overall performance for multiple language identification system is computed from the pooled set of all trial scores, that is the mean of the individual language result.
Chapter 3

Related Works

3.1 Overview

Although the research and development of LID systems began in early 1970s [33], the progress of LID was slow in the first few years due to the lack of public-domain multilingual speech corpus. When the OGI-TS corpus (Oregon Graduate Institute Multilanguage Telephone Speech Corpus) [34] was public available in early 1990s, more and more researchers began to use this corpus and apply different methods on the LID task. With the trend of globalization, automatic language identification being an indispensable part of multilingual speech and language processing system received renewed researcher’s attention. This leads to National Institute of Standards and Technology (NIST) hosting a Language Recognition Evaluation (LRE) competition which resulted in new efforts in this field. The first formal evaluation was held in 1996, then 2003 and 2005 [51]. NIST will continue to organize such competition and aims to establish a new baseline of performance for language recognition of conversational telephone speech and to lay the groundwork for further research efforts in the field.

In the following sections, we first introduce the NIST LRE as all the experiment results reported in this report are based on the NIST LRE 1996 and 2003 evaluation sets. We will also discuss two novel feature extraction methods that are successfully used in LID. Finally we will describe some of the state-of-the-art LID systems which fall in two main categories: phonotactic based and cepstral based LID systems.
3.1.1 1996 and 2003 NIST LRE

The systems described in this report are evaluated with the NIST 1996 and 2003 language recognition evaluations (LRE) corpus. The goal of the NIST LRE was to quantify performance of language identification systems for conversational telephone speech using uniform evaluation procedures [51]. Currently, NIST had already held 3 evaluations on 1996, 2003 and 2005. The task in all evaluations was to recognize the language being spoken in speech utterances of three durations (30s, 10s and 3s). The goal of 1996 and 2003 evaluations was to recognize the language being spoken in speech utterances from a set of 12 target languages: English, Mandarin, Spanish, Arabic, Farsi, French, German, Hindi, Japanese, Korean, Tamil, Vietnamese [51].

The 1996 LRE test set consists of approximately 1500 messages for each duration: 480 for English, 160 each for Mandarin and Spanish, and 80 for each of the other nine languages. English messages were obtained from both the CallFriend corpus (160) and other English corpora (320). Other language messages were from CallFriend corpus. The test messages for English, Mandarin and Spanish including two dialects.

The 2003 LRE evaluation set contains 1280 messages for each of the three durations, with the message breakdown as follows: 80 messages from the CallFriend Corpus for each of the 12 target languages, 80 English messages from the Switchboard-1 corpus, 80 English messages from the Switchboard-cellular corpus, 80 Japanese messages from the CallHome corpus, and 80 Russian messages from the CallFriend corpus. The Russian messages are not one of the target language, it is intentionally included to make the evaluation tasks more difficult than in 1996 LRE.

3.2 Feature Extraction

Most of the features used in LID system are similar to the conventional features used in speech recognition. For example, standard MFCC [37], [31] and LPC [46] features are commonly used in LID tasks. One of the recent improvement in language recognition accuracy is due to the discovery of better feature set for language identification. The following section will introduce two new features introduced to the LID systems: SDC and TRAP.
3.2.1 Shifted Delta Cepstra (SDC)

One improved feature is the shift delta cepstral (SDC) coefficients, an extension of delta-cepstral coefficients [30]. SDC coefficients are calculated as shown in Figure 3.1.

![Figure 3.1: Shifted delta cepstral coefficients](image)

SDC coefficients are based upon four parameters, typically written as n-d-p-k. The parameter $n$ is the number of cepstral coefficients used to computed MFCC at each frame; i.e., one frame is represented by a coefficient vector:

$$
c(t) = [c_0 \ldots c_i \ldots c_{n-1}] \quad (3.1)
$$

where $c_i$ stands for the MFCC coefficient, $t$ stands for the coefficient index. The parameter $d$ determines the spread over which deltas are calculated. The parameter $p$ determines the gaps between successive delta computations, and the parameter $k$ is the number of blocks whose delta coefficients are concatenated to form the final feature vector. I.e., for a given time, $t$, we first obtain

$$
\Delta c(t, i) = c(t + i \times p + d) - c(t + i \times p - d) \quad (3.2)
$$
as an intermediate calculation, where $0 \leq t < k$. The SDC coefficients is a vector of these $k$ coefficients.

$$SDC(t) = [\Delta c(t, 0) \Delta c(t, 1)] \ldots \Delta c(t, k - 1)] \quad (3.3)$$

Hence, $k \times n$ parameters are used for each SDC feature vector, as compared with $2 \times n$ for conventional cepstra and delta-cepstra feature vectors.

The SDC feature captures the long time cepstral dynamics, it compensates the limitation of the traditional short time derivation of the cepstra features. MIT Lincoln lab reported their performance using GMM-based LID systems with SDC coefficients are comparable to their P-PRLM system [28]. Prior to the use of SDC coefficients, GMM-based language recognition was less accurate than alternate approaches [22].

### 3.2.2 Long-time TempoRAI Patterns (TRAP)

The long-time TempoRAI Patterns (TRAPs) of spectral energies was successfully applied for speech recognition [40], it is also introduced in the LID area. The research group from from Brno University of Technology (BUT) reported their P-PRLM LID system using TRAP feature in phoneme tokenization, their P-PRLM system had achieved favorable performance as an individual system in literature [42].

As shown in Figure 3.2, the conventional feature extraction method divides the incoming signal into short-term frames (typically 10 to 20 ms) which are equally spaced in time, spectral coefficients are calculated from each frame, so each frame is represented as a feature vector. Each feature vector typically characterizes the spectral properties of the speech frame. In this way, the input continues speech signal is converted to a matrix of spectral coefficients.

Figure 3.3 shows the process of TRAP based feature extraction. The TRAP features are based on narrow band spectrum with long time context. It divides the speech signal into sub-bands (called critical bands) according to the frequency, and capture the spectral information in long time span based on specified window size. The spectral coefficient vector from each sub-band is firstly transformed with DCT, and then DCT transformed vectors from all sub-bands are combined into one long vector. Follow this procedure, the input speech data is represented by a matrix of these long DCT vectors where each
Figure 3.2: Conventional feature extraction

DCT vector characterizes one window. The TRAP feature vectors can be obtained by applying dimensional reduction technique on this matrix. Many dimensional reduction techniques can be adopted, for example, principal component analysis (PCA).

The main issues involved in designing TRAP feature includes:

- 1) The definition of frequency sub-bands. With more and narrower sub-bands, frequency-localized degradation is reduced, however there is a corresponding loss in spectral information. And with exceedingly narrow bands, the feature would not be useful as it has poor discrimination power between the linguistic classes because of reduced spectral information in each sub-band. With border sub-bands, the spectral information maybe too general that loss the specific characters for certain bands.

- 2) The features to be used in each sub-band. All the standard speech features can be used, for example, MFCC, PLP etc.

- 3) The window size. The number of the long DCT coefficient vectors grows with the decreasing of the size of the window, so short window increase the computational
3.3 Five state-of-the-art LID systems

In this section, we will discuss five state-of-the-art LID systems. They can mainly be classified into two categories: phonotactic based and cepstral based. The cepstral based LID systems model the spectral feature vectors directly, while the phonotactic based LID systems convert these features into intermediate representation for processing in the form of linguistically-defined units.

Table 2.1 and Table 2.2 show the EER of five LID systems on NIST LRE 1996 and 2003 evaluation data [28] [42] [50]. We will briefly describe these LID systems in the following sections.

complexity. On the other hand, long window may not detail enough to capture the cepstral variation. The usual size of the TRAP feature used in ASR in 1 second [41].
### 3.3.1 Phonotactic based LID system

The phonotactic based system is the most widespread approach to language identification. It is based on the likelihood of phoneme sequences extracted from the test signal for each evaluated language. It is believed that the statistical characteristics of the phoneme sequence contains the discriminate information for languages. In the phonotactic based system, the speech signal is first converted into a sequence of meaningful discrete sub-word units (tokens). The derived sequences of tokens are then modelled to distinguish the languages.

#### 3.3.1.1 PRLM

A PRLM system is a language-independent phoneme recognition followed by a language-dependent language modeling module [36]. Figure 3.3 represents the diagram of the standard PRLM LID system.

Assume there are \( n \) target languages: \( l_1, l_2, ..., l_i, ..., l_n \), a PRLM system consists of one phoneme tokenizer and \( n \) language modes. These language models are trained from the speech data in that specific target language. For example, to train a English language model, a collection of speech data in English are past to the phoneme tokenizer, a set

---

**Table 3.1: EER of NIST 1996 LRE**

<table>
<thead>
<tr>
<th>System/Duration</th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIT P-PRLM</td>
<td>5.6</td>
<td>11.9</td>
<td>24.6</td>
</tr>
<tr>
<td>MIT GMM</td>
<td>5.1</td>
<td>8.2</td>
<td>16.4</td>
</tr>
<tr>
<td>MIT Cepstral SVM</td>
<td>4.2</td>
<td>11.7</td>
<td>24</td>
</tr>
<tr>
<td>I2R BOS</td>
<td>4.87</td>
<td>11.18</td>
<td>22.38</td>
</tr>
<tr>
<td>LIMSI P-PRLM</td>
<td>3.2</td>
<td>7</td>
<td>15.6</td>
</tr>
<tr>
<td>BUT P-PRLM</td>
<td>1.48</td>
<td>5.66</td>
<td>15.83</td>
</tr>
</tbody>
</table>

**Table 3.2: EER of NIST 2003 LRE**

<table>
<thead>
<tr>
<th>System/Duration</th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIT P-PRLM</td>
<td>6.6</td>
<td>14.3</td>
<td>25.5</td>
</tr>
<tr>
<td>MIT GMM</td>
<td>4.8</td>
<td>9.8</td>
<td>19.8</td>
</tr>
<tr>
<td>MIT Cepstral SVM</td>
<td>6.1</td>
<td>16.4</td>
<td>28.2</td>
</tr>
<tr>
<td>I2R BOS</td>
<td>6.33</td>
<td>13.35</td>
<td>24.3</td>
</tr>
<tr>
<td>LIMSI P-PRLM</td>
<td>4</td>
<td>8.3</td>
<td>18.3</td>
</tr>
<tr>
<td>BUT P-PRLM</td>
<td>2.42</td>
<td>8.08</td>
<td>19.08</td>
</tr>
</tbody>
</table>
of phoneme sequences are generated. These phoneme sequences are used to train the 'English' language model.

Given an unknown speech segment, it is firstly converted into a feature vector in the feature extraction process. This feature vector is then feed into the phoneme tokenizer. The output of the phoneme tokenizer is a phoneme sequence with \( p \) phonemes: \( s = s_1, s_2, ..., s_p \). The sequence \( s \) is compared with each of the \( n \) language models that are trained from their corresponding target languages. The probability distribution for each language model is estimated, we call them the language scores. Hence \( n \) language scores are calculated from each language models. The speech segment is classified as the language that gives the highest language score.

![PRLM diagram](image)

Figure 3.4: PRLM diagram

The phonotactic n-gram language modeling is widely adopted in PRLM system [22], [36]. More specifically, the unigram, bigram, and trigram model and their combinations are most popular in LID systems.

In the derivation of the n-gram model, the statistical independence assumption of phoneme elements is made. For a unigram model each phoneme element is assumed to be statistically independent of all other phoneme elements. That is:

\[
P(s|l_i) = P(s_1, s_2, ..., s_p|l_i) = \prod_{k=1}^{p} (s_k|l_i)
\] (3.4)

where \( l_i \) is the target language, \( s_1, s_2, ..., s_p \) stands for the \( p \) phoneme sequence that generated from the phoneme tokenizer.
A bigram model assumes each phoneme is dependent on only the phoneme immediately preceding it.

\[ P(s|l_i) = P(s_1|l_i) \prod_{k=2}^{p} (s_k|s_{k-1}, l_i) \]  

(3.5)

Similarly, a trigram model assumes each phoneme is statistically dependent on the two preceding phonemes.

\[ P(s|l_i) = P(s_1|l_i)P(s_2|l_i) \prod_{k=3}^{p} (s_k|s_{k-1}, s_{k-2}, l_i) \]  

(3.6)

### 3.3.1.2 P-PRLM

Zissman extends the idea of PRLM to parallel phoneme tokenizer [37] as it is believed that different phoneme tokenizer will present the language information from different angle. This system is called the P-PRLM system and is illustrated in Figure 3.4.

The P-PRLM system consists of \( d \) \((d > 1)\) phoneme tokenizers instead of only one in the PRLM system, and it should have \( d \times n \) language models. In the training process, for every phoneme tokenizer, we need to train \( n \) language models following the same process as described in the PRLM system.

In the test process, an unknown speech segment now needs to pass through all the \( d \) phoneme tokenizers and therefore \( d \) phoneme sequences are generated. The phoneme sequence which is generated from the specific tokenizer is estimated by the \( n \) language models that trained from that tokenizer. Hence altogether there are \( d \times n \) language scores been generated. Compared with PRLM system, more complicate classifiers can be adopted to made final decision from those language scores.

Although the computation effort of the P-PRLM system is several times more than PRLM system, much better accuracy is achieved by P-PRLM system. Nowadays, the P-PRLM system became one of the most popular method used in LID task.

MIT Lincoln lab’s P-PRLM system (MIT P-PRLM) [28] uses six phoneme recognizers trained from six OGI-TS [52] languages (English, German, Hindi, Japanese, Mandarin and Spanish). The phoneme recognizers are trained with null-grammar, three-state, six-mixture HMMs [28]. The input feature is a 37-dimensional feature vector derived from HTK3.1 MFCC coefficients. Phoneme sequences produced by the tokenizers are used
to compute gender independent language models for each of the target languages. To better model the speech segments in different length, (as there are three types of duration test segments in the NIST LRE: 30 seconds, 10 seconds, 3 seconds) duration-dependent Gaussian classifiers are trained from the P-PRLM score vectors to produce language scores.

Another successful P-PRLM system is developed by a research group in France, LIMSI [50]. They improved the phonotactic n-gram language modeling by taking the summation over the phoneme sequences present in the phone lattice instead of using just the top one choice [50]. Three phone recognizers for Arabic, American English, and Spanish were used in their system. Each phone model is a tied-state HMM with 32 Gaussians per state. The acoustic feature vector has 39 components comprising of 12 PLP cepstrum coefficients and the log energy, along with their first and second order derivatives.

Brno University of Technology (BUT) also reported their P-PRLM system (BUT P-PRLM) which uses 4 phone recognizers trained from SpeechDat-E corpus [53], they are Czech, Hungarian, Polish, and Russian [42]. Their feature extraction module makes use of long temporal context: TRAPs, as discussed previously. Trigram language model was
used to capture phonotactic statistics of each language. So far, this P-PRLM system generated best performance on NIST 1996 and 2003 data set as shown in table 3.1 and 3.2.

3.3.1.3 Bag-of-sounds

The Bag-of-Sounds(I2R BOS) is a novel concept of LID proposed by Ma and Li [31]. As illustrated in Figure 3.5, the system assumes that all languages in the world can be covered by a universal collection of acoustic tokens or sound tokens, this is called the bag-of-sounds. It’s concept is analogous to the concept of the bag-of-words [43] in the text processing system. With these acoustic tokens, any utterance can be represented as sequences of these tokens, and hence any classification method can be applied to distinguish languages.

![Figure 3.6: BOS system](image)

The system has three components: the universal sound recognizer, the bag-of-sounds representation and the bag-of-sounds classifier.

**Universal sound recognizer:** The idea is to use one recognizer to tokenize speech signal of all candidate languages into sound sequences. To achieve this, the key problem is to find the set of acoustic tokens to model all languages. An ideal universal sound recognizer should be able to cover all sounds that may appear in every languages or dialects in the world, but it is not practical for implementation. One approach to find the acoustic tokens is to use unsupervised learning procedure to circumvent the need for phonetic transcription. More specifically, with training data from many languages,
we can use unsupervised training program to cluster the sound patterns into specified number of categories, these categories can be used as the basic sound tokens to train the sound recognizer. Another approach is to use a super set of phonemes of several languages as the universal sound tokens. The second approach is adopted in the BOS system described in this thesis.

The bag-of-sound representation module: Given the input speech signal, the universal sound recognizer will generate a sequence of sound tokens,

\[ s_0, s_1, s_2, \ldots, s_i, \ldots, s_n \]  

(3.7)

where \( s_i \) is an unique sound token, \( n \) is the number of the sound token in the sound sequence, \( i \) denote the index of the sound token. For simplicity, here we assume a sound token is a phoneme/phone.

To capture the local phonotactics, a phoneme sequence is converted into bi-phone count vector. i.e.

\[ c = [c_{0,0}, c_{0,1}, \ldots, c_{0,M}, c_{1,0}, c_{1,1}, \ldots, c_{1,M}, \ldots, c_{M,M}]^T \]  

(3.8)

where \( c_{i,j} \) denotes the number of occurrence of bi-phone unit \( s_i s_j \) in the given speech segment, \( M \) is the number of unique phones in the universal sound inventory, then there are possibly \( N \) bi-phone entries, where:

\[ N = M \times M \]  

(3.9)

Thus a speech segment is represented by a \( N \) dimensional vector of sound frequency statistics.

As speech segments usually differ in length, the numbers of bi-phone occurrence are not comparable between two segments, normalization mechanism is applied. For each element in (3.8), we use the following equation to transform a bi-phone count to a bi-phone occurrence probability:

\[ w_{i,j} = \frac{(1 - \varepsilon_{i,j})c_{i,j}}{\sum c_{i,j}} \]  

(3.10)

where \( c_{i,j} \) is the number of times bi-phone token \( s_is_j \) occurs in given speech session, \( \sum c_{i,j} \) is the total number of bi-phone tokens occurs in the speech segment, and \( \varepsilon_{i,j} \) is the normalized entropy of bi-phone token \( s_is_j \) in the multilingual speech corpus. It is
calculated from the training corpus. Suppose there are \( R \) speech sessions in the speech corpus \( \Gamma \), we have

\[
\varepsilon_{i,j} = -\frac{1}{\log R} \sum_{j=1}^{R} \frac{c_{i,j}}{t_{i,j}} \log \frac{c_{i,j}}{t_{i,j}}
\]

(3.11)

where \( t_{i,j} = \sum_j c_{i,j} \) is the total number of times the bi-phone \( s_is_j \) occurs in speech corpus \( \Gamma \). From this definition, we have

\[
0 \leq \varepsilon_{i,j} \leq 1
\]

(3.12)

where the value of \( \varepsilon_{i,j} \) close to 1 indicates the bi-phone token is distributed across many languages, this means the bi-phone token is not very useful to distinguish the languages. While a value of \( \varepsilon_{i,j} \) close to zero means that the bi-phone token is present only in a few specific languages, this means this bi-phone token contains more discriminative information for languages. This explains why in (3.10), we use \( 1 - \varepsilon_{i,j} \) as the weighting factor.

In this way, a speech session is represented by a *bag-of-sounds* vector \( r \):

\[
r = [w_1 \ w_2 \ ... \ w_i \ ... \ w_N]^t
\]

(3.13)

**Bag-of-sounds classifier**: Using the *bag-of-sounds* representation described previously, for the training speech corpus \( \Gamma \), suppose there are \( R \) speech sessions, every speech session can be represented as a vector (3.15), then the whole speech corpus became a matrix \( W \) with \( N \times R \) elements. Thus a language identification task became a vector classification problem. A Support Vector Machine (SVM) is used to partition the high dimensional vector space [62].

Support vector machine is a binary classifier, in the system, we trained such a binary classifier for each language pair. For a training set of \( ((x_1, y_1), \ldots (x_i, y_i), \ldots, (x_R, y_R)) \), the binary classifier is learned:

\[
c : x \rightarrow y, y = \text{sign}(\lambda x + b)
\]

(3.14)

The final LID decision can be generated by combining the output of these SVM classifiers.
3.3.2 Cepstral based LID system

Although the phonotactic based system has good accuracy, such approach may be impractical as it involves building of multilingual phoneme recognition system which requires extensive knowledge about the acoustic-phonetic, lexical, and linguistic rules for each of the languages of interest. In addition, it is a time and labor intensive task to obtain enough speech data with transcription to train a phoneme tokenizer. Thirdly the computation complexity of having the unknown utterance passing through several phoneme tokenizers make this LID system not efficient enough for real time processing.

An alternative to phonotactic based system is the cepstral based system. The cepstral based system is motivated by the observation that humans can often identify the language of an utterance even with little working linguistic knowledge of that language. This phenomena indicates that the listener in the absence of higher level knowledge of a language, he presumably relies on lower level constraints. Inspired by this observation, the cepstral based LID system focus on the modeling of the cepstral level.

Different from the phonotactic based LID system, the cepstral based LID system train the discriminative classifiers on the cepstral features level, there is no phoneme recognizer needed. This make the cepstral based LID system more practical for real time LID applications. In this subsection, we will describe two types of cepstral based LID system: Cepstral GMM LID and Cepstral SVM LID.

3.3.2.1 Cepstral GMM

The most widely used cepstral based system is use Gaussian mixture model to classify sounds of different languages [30] [37].

Under the GMM assumption, each feature vector \( \mathbf{v}_t \) at frame time \( t \) is assumed to be drawn randomly according to a probability density that is a weighted sum of multi-variate Gaussian densities:

\[
p(\mathbf{v}_t|\lambda) = \sum_{k=1}^{N} p_k b_k(\mathbf{v}_t)
\]

(3.15)

where \( \lambda \) is the set of model parameters

\[
\lambda = \{ p_k, \mu_k, \Sigma_k \}
\]

(3.16)
Chapter 3. Related Works

$b_k$ is the multi-variate Gaussian densities defined by the means $\mu_k$ and variances $\Sigma_k$

$$b_k \sim N(\mu_k, \Sigma_k)$$  \hspace{1cm} (3.17)

$k$ is the mixture index ($1 \leq k \leq n$), the $p_k$ are the mixture weights constrained such that $\sum_{k=1}^{n} p_k = 1$, where $n$ is the total number of mixtures.

During recognition, an unknown speech utterance is firstly converted from the digitized waveform into feature vectors, and then the feature vectors are evaluated by each Gaussian mixture models of the target languages, the log likelihood scores for each target languages are calculated. The unknown speech is classified to the language which generating the highest log likelihood score.

Lincoln lab’s GMM system (MIT GMM) uses Shifted Delta Cepstra (SDC) feature vectors as input [28]. In their system, a single universal background model (UBM) with 2048 mixtures is firstly trained from the entire train set, then language-dependent model for each target language is adapted from the UBM using the language specific portions of the data. Consider about the duration of the test speech segment, three sets of duration-dependent Gaussian classifiers are trained from the training data with specific duration (i.e. 30 seconds, 10 seconds and 3 seconds). According to the length of the test speech segment, the test segment is evaluated on one set of these duration based GMM classifiers and language scores are generated, and these language dependent scores are converted to log likelihood ratios by subtracting the scores generated from the UBM model. These log likelihood ratios are compared to make final language identification decision.

As shown in table 3.1 and 3.2, MIT Lincoln lab’s GMM LID system achieves better accuracy than their P-PRLM system.

### 3.3.2.2 Support Vector Machines

MIT Lincoln lab proposed their newest LID system using a purely acoustic approach (MIT Cepstral SVM) [28]. For input features, they use shifted delta cepstral coefficients (SDC). For language classifier, they use a support vector machine (SVM) as discriminative classifier, this system is originally developed for speaker recognition [70], it is also shown good performance when applied in the LID task.
Support vector machine is a popular technique for pattern classification. An SVM \[55\] is a two class classifier constructed from sums of a kernel function \( K(.,.) \)

\[
f(x) = \sum_{i=1}^{N} \alpha_i t_i K(x, x_i) + d \tag{3.18}
\]

where the \( t_i \) is the ideal output, it is either 1 or -1, depending upon whether the corresponding support vector is in class 0 or class 1. \( \sum_{i=1}^{N} \alpha_i t_i = 0, \) and \( \alpha_i > 0. \) The vectors \( x_i \) are support vectors and obtained from the training set by an optimization process.

The kernel \( K(.,.) \) is constrained to have certain properties (the Mercer condition), so that \( K(.,.) \) can be expressed as

\[
K(x, y) = b(x)^t b(y) \tag{3.19}
\]

where \( b(x) \) is a mapping from the input space (where \( x \) lives) to a possibly infinite dimensional space.

To adopt SVM to a classification task involving speech is not straightforward as speech classification problems involve sequences of feature vectors, i.e., a speech utterance is often represented by frame level feature vectors, the length of the utterance may not same. While the typical SVM is used on vectors that have same dimension. We need to find a way of taking a sequence of input feature vectors from an utterance and computing the SVM output. One way of handling this situation is to assume that the kernel in the SVM takes sequences as inputs; i.e., we can calculate \( K(x_i, y_j) \) for two input sequences \( x_i \) and \( y_j. \) We call this a sequence kernel method.

The Lincoln lab proposed a Generalized Linear Discriminant Sequence (GLDS) kernel based upon comparing sequences of speech feature vectors \[57\] \[70\]. The following paragraphs summarize the GLDS kernel.

**Generalized Linear Discriminant Scoring**

We can represent our LID as a two class problems, i.e., target and nontarget language. If \( \omega \) is a random variable representing the hypothesis, then \( \omega = 1 \) means the target present and \( \omega = 0 \) means the target not present.

A score is calculated from a sequence of observations \( y_1, ..., y_n \) extracted from the speech input. The scoring function is based on the output of a generalized linear discriminant function \[56\] of the form

\[
g(y) = w^t b(y) \tag{3.20}
\]
where $w$ is the vector of classifier parameters (model) and $b$ is an expansion of the input space into a vector of scalar functions. An example is

$$b(y) = [b_1(y) b_2(y) \ldots b_n(y)]^t$$  \hspace{1cm} (3.21)

where $b_i$ is a mapping from $\mathbb{R}^m$ to $\mathbb{R}$. Commonly used generalized linear discriminants are polynomials [71] and radial basis functions [72].

If the classifier is trained with a mean-squared error training criterion and ideal outputs of 1 for $\omega = 1$ and 0 for $\omega = 0$, then $g(y)$ will approximate the a posteriori probability $p(\omega = 1|y)$ [72]. Assuming independence of the observations, we can then find the probability of the entire sequence, $p(y, \ldots, y_n|\omega = 1)$ as:

$$p(y_1, \ldots, y_n|\omega) = \prod_{i=1}^{n} p(y_i|\omega) = \prod_{i=1}^{n} \frac{p(\omega|y_i)p(y_i)}{p(\omega)}$$  \hspace{1cm} (3.22)

For the purpose of classification, we can discard $p(y_i)$. We take the logarithm of both sides to get the discriminant function

$$d(y^n_i|\omega) = \sum_{i=1}^{n} \log\left(\frac{p(\omega|y_i)}{p(\omega)}\right)$$  \hspace{1cm} (3.23)

where we $y^n_i$ denotes the sequence of vectors $y_1, \ldots, y_n$. We use Taylor series of $\log(x) \approx x - 1$ to obtain the final discriminant function

$$d(y^n_1|\omega) = \frac{1}{n} \sum_{i=1}^{n} \frac{p(\omega|y_i)}{p(\omega)}$$  \hspace{1cm} (3.24)

Note that we have discarded the -1 in this function and normalized by the number of frames since these changes will not affect the classification decision.

Now assume we have $g(y) \approx p(\omega = 1|y)$, we call the vector $w$ the target model. Substituting in the generalized linear discriminant approximation $g(y)$ in (3.20) gives:

$$d(y^n_1|\omega = 1) = \frac{1}{n} \sum_{i=1}^{n} \frac{w^t b(y_i)}{p(\omega = 1)} = \frac{1}{np(\omega = 1)} w^t \left( \sum_{i=1}^{n} b(y_i) \right) = \frac{1}{p(\omega = 1)} w^t \bar{b}_y$$  \hspace{1cm} (3.25)

where we define the mapping $y^n_1 \rightarrow \bar{b}_y$ as

$$y^n_1 \rightarrow \frac{1}{n} \sum_{i=1}^{n} b(y_i)$$  \hspace{1cm} (3.26)
That is, for a sequence of input vectors \( y_1, \ldots, y_n \) and a target model \( \mathbf{w} \), we construct \( \mathbf{b}_y \) using (3.26). We then score using the target model, i.e., score = \( \mathbf{w}^T \mathbf{b}_y \).

**Generalized Linear Discriminant Classifier Training**

We next review how to train the classifier. Let \( \mathbf{w} \) be the desired target model. The training problem is

\[
\mathbf{w}^* = \arg \min_{\mathbf{w}} E[(\mathbf{w}^T \mathbf{b}(\mathbf{x}) - \omega)^2] \tag{3.27}
\]

where \( E \) is the expectation, \( \mathbf{x} \) denotes the training vectors. This criterion can be approximated using the training set as

\[
\mathbf{w}^* = \arg \min_{\mathbf{w}} \left[ \sum_{i=1}^{N_{tgt}} |\mathbf{w}^T \mathbf{b}(\mathbf{x}_i) - 1|^2 + \sum_{i=1}^{N_{non}} |\mathbf{w}^T \mathbf{b}(\mathbf{z}_i)|^2 \right] \tag{3.28}
\]

Here, the target training data are \( \mathbf{x}_1, \ldots, \mathbf{x}_{N_{tgt}} \) and the non-target data are \( \mathbf{z}_1, \ldots, \mathbf{z}_{N_{non}} \).

We define \( \mathbf{M}_{tgt} \) as the matrix whose rows are the expansion of the target’s data, i.e.

\[
\mathbf{M}_{tgt} = \begin{pmatrix}
\mathbf{b}(\mathbf{x}_1)^T \\
\mathbf{b}(\mathbf{x}_2)^T \\
\vdots \\
\mathbf{b}(\mathbf{x}_{N_{tgt}})^T
\end{pmatrix} \tag{3.29}
\]

Similarly we can define a matrix \( \mathbf{M}_{non} \) for the non-target data and we define

\[
\mathbf{M} = \begin{pmatrix}
\mathbf{M}_{tgt} \\
\mathbf{M}_{non}
\end{pmatrix} \tag{3.30}
\]

The problem then becomes

\[
\mathbf{w}^* = \arg \min_{\mathbf{w}} ||\mathbf{M}\mathbf{w} - \mathbf{o}||_2 \tag{3.31}
\]

where \( \mathbf{o} \) is the vector consisting of \( N_{tgt} \) ones followed by \( N_{non} \) zeros. Now the problem can be solved using

\[
\mathbf{M}'\mathbf{M}\mathbf{w} = \mathbf{M}'\mathbf{o} = \mathbf{M}_{tgt}'\mathbf{1} + \mathbf{M}_{non}'\mathbf{0} = \mathbf{M}_{tgt}'\mathbf{1} \tag{3.32}
\]

where \( \mathbf{1} \) and \( \mathbf{0} \) are the vectors of all ones and all zeros, respectively. If we define \( \mathbf{R} = \mathbf{M}'\mathbf{M} \) and solve \( \mathbf{w} \)

\[
\mathbf{w} = R^{-1}\mathbf{M}_{tgt}'\mathbf{1} \tag{3.33}
\]
Chapter 3. Related Works

Generalized Linear Discriminant Sequence Kernel

Combine the target model from (3.33) with the scoring equation from (3.25) to obtain the classifier score

$$score = \frac{1}{p(\omega = 1)} \vec{b}_y w = \frac{1}{p(\omega = 1)} \vec{b}_y R^{-1} M_{tgt} 1$$  \hspace{1cm} (3.34)$$

Now \(p(\omega = 1) = \frac{N_{tgt}}{N_{tgt} + N_{non}}\), so that the above score becomes

$$score = \vec{b}_x^t R \vec{b}_y$$  \hspace{1cm} (3.35)$$

where \(\vec{b}_x\) is \(\frac{1}{N_{tgt}} M_{tgt} 1\), this is exactly same as mapping in (3.26), and \(R\) is \(\frac{1}{N_{tgt} + N_{non}} R\). So the GLDS kernel is defined as:

$$K_{GLDS}(x^n_1, y^m_1) = \vec{b}_x^t R \vec{b}_y$$  \hspace{1cm} (3.36)$$

Given two sequences of speech feature vectors \(x^n_1\) and \(y^m_1\), we compare them by mapping \(x^n_1 \rightarrow \vec{b}_x\) and \(y^m_1 \rightarrow \vec{b}_y\) and computing GLDS kernel. The value of the \(K_{GLDS}(x^n_1, y^m_1)\) can be interpreted as scoring using a generalized linear discriminant on the sequence \(y^m_1\) with the MSE model trained from feature vectors \(x^n_1\).

This kernel has two advantages. First, since we are using an explicit expansion \(b()\), we can simplify to a single model as in (3.33). Second, the kernel produces input speech sequences.

In their LID system, the SVM uses GLDS kernel with an expansion into feature space using a monomial basis. All monomials up to degree 3 are used. To simplify the training, a diagonal approximation to the kernel inner product matrix is used. The feature vector sequence from each conversation side is divided into 5 equal sections, and each section is used to produce an average feature space expansion. After finding the average feature space expansion vectors for all languages, a SVM is used to produce language models using the GLDS kernel. Language model scores for each test utterance are obtained by computing the inner product between the language model and the average expansion of the utterance.
Chapter 4

Proposed LID System

4.1 Overview

The research in the first year focus on exploring the acoustic, prosodic and phonotactic features in LID. We developed five LID systems that concentrate on different level of discriminative features, including spectrum, duration, pitch, n-gram phonotactic, and bag-of-sounds features. As illustrates in Figure 4.1.

At acoustic level, we extract the Shift-delta-cepstra (SDC) feature from the conventional MFCC features. The use of SDC coefficients aims to incorporate additional
temporal information about the speech into the feature vectors for more discriminative information. The Gaussian mixture models (GMMs) are used to model SDC feature vectors, these models are used to evaluate the unknown speech segments and output the log likelihood scores.

At prosodic level, we explored two prosodic information: one is the duration of the phoneme, another one is the pitch information. For both system GMM models are trained from these prosodic features and used to evaluate the unknown speech.

At phonotactic level, we use two different approaches to capture the phonotactic information. The first one follows the common P-PRLM architecture. The other system implements the Bag-of-sounds approach proposed from our lab. It introduces a single universal sound tokenizer which segments speech signal into phone sequences, then extracts bi-phone statistics and projects the statistics into a high dimensional space for SVM to carry out discrimination.

In the following sections, firstly, we will introduce the development data, then describe five individual systems and the experiment results.

4.2 Language Identification Corpora

The NIST 1996 and 2003 LRE database were used to evaluate the system described in this report. Our development data was from CallFriend corpus [58]. We use the same 12 languages (Arabic, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Spanish, Tamil, Vietnamese and English) and 3 dialects (Dialects of English, Mandarin and Spanish) as the target languages specified in the NIST 1996 LRE. The training data consisted of 20 complete conversations (normally 30 minutes) for each of the target languages. In CallFriend corpus, data for each language were grouped into 3 parts: “train”, “devtest” and “evaltest”. As some of the testing data in NIST LRE are extract from the “evaltest” set, we only using “train” and “devtest” as our development data. All the development and test data are pre-processed by a speech activity detection program to remove silence and chunk into smaller speech segments.

In the development process, we treat the dialects of English, Mandarin and Spanish as different languages. Therefore, there are 15 languages in the training process. For our
results to be comparable with other reports in the literature, in the test process, we only measure the LID performance of the primary languages by grouping the dialect labels into their respective primary language.

![Figure 4.2: Five LID systems](image)

As illustrated in Figure 4.2, the development of these LID systems consists of four process modules: feature extraction, front-end discriminative feature representation, back-end classifiers and final decision classifiers. The feature extraction and final decision classifiers used in all the 5 systems are same, they all use MFCC features. In the following section, we will focus on describing the other two modules in each system.

### 4.3 LID on Acoustic, Prosodic and Phonotactic Features

#### 4.3.1 N-gram LM in P-PRLM

Following the P-PRLM formulation as in [22], seven phoneme tokenizers were used in our system: English, Korean, Mandarin, Japanese, Hindi, Spanish and German. English phonemes were trained from IIR-LID [60] database. Korean phonemes were trained from
LDC Korean corpus (LDC2003S03) [59]. Mandarin phonemes were trained from MAT corpus [61]. Other phonemes were trained from OGI-TS corpus [52].

The speech signal was divided into 10 ms frames, for each frame a 39-dimensional MFCC feature vector is extracted. Each feature vector consisted of 12 MFCC coefficients, the log energy and their first and second order derivatives. Utterance based cepstral mean subtraction was applied to the MFCC features to remove channel distortion. Each phoneme in the languages was modeled with a 3-state HMM. The English, Korean and Mandarin states had 32 mixtures each, while others had only 6 mixtures due to the reduced availability of training data.

We define the front-end feature extraction in P-PRLM system as the process of converting the speech signal into phoneme sequences. Based on the phoneme sequence from each tokenizer, we counted the occurrences of n-grams: i.e., the subsequences of n symbols (phonemes, in this case). Training was performed by accumulating a set of n-gram histograms, one per language, under the assumption that different languages will have different n-gram histograms. Here we trained up to 3-gram phoneme language model (LM) for each tokenizer-target language pair, as there were 15 target languages and 7 phoneme tokenizers, all together there were 105 = 15 × 7 LMs are trained. We call these 105 language models as the back-end classifiers for the P-PRLM system.

We then use interpolated n-gram language models to approximate the n-gram distribution as the weighted sum of the probabilities of the n-gram, the (n-1)-gram, etc. Here the interpolation of 3-gram model is:

\[ \hat{P}(w_t | w_{t-1}) = \alpha_3 P(w_t | w_{t-1}w_{t-2}) + \alpha_2 P(w_t | w_{t-1}) + \alpha_1 P(w_t) + \alpha_0 P_0 \]  

(4.1)

where \( w_t, w_{t-1} \) and \( w_{t-2} \) are consecutive symbols observed in the phone stream. The \( P \)'s are ratios of counts observed in the training data, e.g.:

\[ P(w_t | w_{t-1}) = \frac{C(w_{t-1}, w_t)}{C(w_{t-1})} \]  

(4.2)

where \( C(w_{t-1}, w_t) \) is the number of times symbol \( w_{t-1} \) is followed by \( w_t \), and \( C(w_{t-1}) \) is the number of occurrences of symbol \( w_{t-1} \). \( P_0 \) is the reciprocal of the number of symbol types. The \( \alpha \)'s are the weighting parameters and \( \sum_{i=0}^{n} \alpha_i = 1 \). The weighting parameters can be estimated iteratively using the Expectation Maximization (EM) algorithm so as
to minimize the perplexity. In this system we selected them according to the practical experimental result.

For each input utterance, we calculated 105 interpolated language likelihood scores from 105 language models described above. Each element of the score vectors are normalized by subtracting the mean of their competing languages. In this way, the development data of 15 languages are represented by a collection of 105-dimensional score vectors. These score vectors of the development data are used to train the final P-PRLM classifier.

The final classifier of the P-PRLM consists of 15 pairs of Gaussian mixture models (GMMs). Specifically, for each target language, we build two GMMs \( \{(m^+, i), (m^-, i)\} \). \( (m^+, i) \) is trained on the score vectors of target language \( i \), called positive model, while \( (m^-, i) \) is trained on those of its competing languages, called negative model.

Given an unknown speech utterance \( O \), we can obtain one phoneme sequences from each of the phoneme tokenizer. From this phoneme sequence, we get 15 language scores. Thus we get 105 language scores from all of the seven phoneme tokenizer. These language scores are combined into a testing vector. We then calculate the likelihood scores of all the 15 pairs of GMMs for this test vector. The confidence of a test utterance for target language \( i \) is given by the likelihood ratio

\[
\lambda_{P-PRLM, i} = \log \left( \frac{p(O|m^+, i)}{p(O|m^-, i)} \right)
\]

where \( p(O|m^+, i) \) is the likelihood score for the the back-end GMM of target language \( i \), and \( p(O|m^-, i) \) is the likelihood score for the negative model of language \( i \). The unknown utterance \( O \) is classified to the language \( k \) which have the highest language confidence

\[
\lambda_{P-PRLM, k} = \max_i (\lambda_{P-PRLM, i})
\]

4.3.2 Bag-of-Sounds (BOS)

The bag-of-sounds method uses a universal sound recognizer to tokenize an utterance into a sound sequence, and then converts the sound sequence into a count vector, known as bag-of-sounds vector [31].

An ideal universal sound recognizer should be able to cover all sounds appear in each languages. To achieve this, all sounds appeared in every language need to be included
in the sound inventory and labelled data for each language is required to train such a recognizer, hence such a system is is not easy to implement. An alternative method is to select a subset of the world’s spoken languages with their phonemes representative for all the candidate languages of interest. We adopted this method in our sound inventory selection. In our system, the universal sound inventory is a combined phoneme set from 6 languages: English, Mandarin, Hindi, Japanese, Spanish and German. The training corpus of these languages are came from the same source as described in our P-PRLM system, which is described in last section. There are a total of \( i = 258 \) phonemes in the system.

Following the bag-of-sounds concept explained in last chapter, the front-end of our BOS is defined as a speech signal being converted to a phoneme sequence \( s \) by the universal sound tokenizer:

\[
s = s_1, s_2, s_3, ..., s_t, ..., s_p
\]

where \( p \) is the number of the phonemes in this sequence, \( s_t \) is one of the phoneme in the 258 phoneme inventory. The phoneme sequence is then represented as a bi-phone vector with \( 66,564 = 258 \times 258 \) elements.

\[
c = [c_{0,0} \ldots c_{0,1} \ldots c_{k,j} \ldots c_{i-1,i} c_{i,i}]^t
\]

where \( c(k,j) \) is defined as the number of co-occurrence of phoneme \( k \) and \( j \) in the given utterance: \( \text{Counts}(k,j) \), normalized by the total number of bi-phone occurrence in the sequence:

\[
c_{k,j} = \frac{\text{Counts}(k,j)}{p - 1}
\]

Hence the development data for each of the 15 languages are represented by a set of the bi-phone vectors described above. We use SVM to discriminate these high dimension bi-phone feature vectors. As SVM is two way classifier, we train pair-wise SVM classifiers for the 15 target languages, resulting in \( 105 = 15 \times 14/2 \) SVM classifiers. The SVM-light tool [62] is used to train the SVM classifiers using linear kernel. The back-end classifiers of the BOS system is consists of these 105 SVM models.

An input training bi-phone vectors is classified by these 105 SVM classifiers to derive a 105-dimensional score vectors. The collection of training score vectors are used to train 15 pairs of final decision classifiers in the same way as it is used in P-PRLM.
Given an unknown speech utterance \( O \), the confidence of a test utterance for target language \( i \) is given by the likelihood ratio

\[
\lambda_{BOS,i} = \log(p(O|m^+, i)/p(O|m^-, i)) \tag{4.8}
\]

where \( p(O|m^+, i) \) is the likelihood score for the the back-end GMM of target language \( i \), and \( p(O|m^-, i) \) is the likelihood score for the corresponding negative model. The unknown utterance \( O \) is classified to the language \( k \) which have the highest language confidence

\[
\lambda_{BOS,k} = \max_i(\lambda_{BOS,i}) \tag{4.9}
\]

### 4.3.3 SDC Feature in GMM

Gaussian mixture models are used to model acoustic characteristics of a language, known as GMM acoustic in [30]. In front-end feature extraction, we use the shifted delta cepstral (SDC) features [28] to capture long time spectral information across successive frames. We adopt the parameter setting 7-3-1-7 as used in [28] to generate the SDC feature. This results in a sequences of features vectors of dimension 49 for each utterance.

Firstly, a Gaussian Mixture model with 2048 mixtures is trained from all the SDC feature vectors of 15 languages, we denote it as the gender independent universal background model (UBM). To model the gender information, we separated the training data from each language into two sets: Male and Female. Then we build two gender dependent UBM models out of these two sets. The goal with a universal background model is that it can be trained once and used as a initial model to do adaptation for all languages. Then Bayesian adaptation is applied to train language dependent models from the UBMs. We first adapt the gender independent UBM towards each target language amounting to 15 language dependent GMMs. Then we further adapt the language dependent GMMs by gender dependent UBMs resulting in 30 gender-language dependent GMMs. In summary, we obtain one gender independent UBM and 15 language dependent GMMs, 2 gender dependent UBMs and 30 gender-language dependent GMMs. We denote these GMMs as back-end classifiers.
The average log-likelihood of a model $\lambda$ given an utterance (represented as a sequence of SDC vectors) $X = x_1, x_2, \ldots x_T$ is computed as

$$L(X|\lambda) = \frac{1}{T} \sum_{t=1}^{T} \log P(x_t|\lambda) \quad (4.10)$$

A likelihood ratio score between the language dependent model, $\lambda_i$, and the likelihood score of corresponding UBM model is used to calculate final score

$$\Lambda(X|i) = L(X|\lambda_i) - L(X|\lambda_{UBM}) \quad (4.11)$$

Follow the equation (4.9), an utterance is evaluated on the 45 GMMs and 3 UBMs to generate 45 language dependent scores in a 45-dimensional vector. The score vectors are then normalized by their respective UBM scores as (4.10).

The collection of training score vectors are used to train a set final decision classifiers in the similar way as it is used in P-PRLM. Specifically, for each target language $i$, we build a pair of GMMs $\{m^+, i\} and \{m^-, i\}$. $m^+, i$ is trained on the score vectors of target language $i$, called positive model, while $m^-, i$ is trained on those of its competing languages, called negative model. In this way, we train 15 pairs of GMMs as the back-end classifiers.

Given an unknown speech utterance $O$, the confidence of a test utterance for target language $i$ is given by the likelihood ratio

$$\lambda_{SDC,i} = \log(p(O|m^+, i)/p(O|m^-, i)) \quad (4.12)$$

where $p(O|m^+, i)$ is the likelihood score for the the back-end GMM of target language $i$, and $p(O|m^-, i)$ is the likelihood score for the corresponding negative model. The unknown utterance $O$ is classified to the language $k$ which have the highest language confidence

$$\lambda_{SDC,k} = \max_i (\lambda_{SDC,i}) \quad (4.13)$$

### 4.3.4 Duration

We believe that the phoneme duration statistics provide language discriminative information and early research has found that duration is useful in the speaker recognition study [63].
In the front-end feature extraction, we tokenize each training utterance by using the same universal sound recognizer as in bag-of-sounds classifier. After tokenization, we obtain duration statistics for each phoneme. The duration feature vector has 3 elements representing the duration of 3 states in a phoneme, i.e., the duration of the 1st, 2nd and 3rd states of the phoneme.

The training process is similar as in GMM method we introduced in last section. For each phoneme, we train a 16-mixture language-independent GMM model as the UBM model using the collection of duration features from all languages. As there are 258 phonemes in the phoneme recognizer, here we get 258 UBMs. For each phoneme in a target language, we adapt the specific UBM of the phoneme towards each target language. We arrive at $3,874 = 258 \times 15$ language dependent phoneme duration models. So the back-end classifiers are consists of 258 UBMs and 3,874 language dependent GMMs.

Given an utterance which represented as a sequence of duration feature vectors

$$
\mathbf{d} = [d_{t,1}, d_{t,2}, d_{t,3} \ldots d_{k,1}, d_{k,2}, d_{k,3}]^T
$$

(4.14)

where $k = 258$ and $d_{t,1}, d_{t,2}, d_{t,3}$ denotes the duration of the first, second and third state of phoneme $t$ respectively. The language score of $d\mathbf{d}$ for the target language $i$ is calculated by multiplying the likelihood ratios from the 258 phoneme duration model pairs.

$$
L(d|\lambda_i) = \prod_{t=1}^{k} \log\left(\frac{p(D|m^+, t, i)}{p(D|m^-, t, i)}\right)
$$

(4.15)

where $p(D|m^+, t, i)$ is the language dependent likelihood score for the phoneme $t$ and language $i$, and $p(O|m^-, t, i)$ is the likelihood score of the UBM model of phoneme $t$ and language $i$. Follow the equation (4.14), the utterance is converted to a score vector of 15 dimensions representing 15 languages.

The collection of the 15 dimensional score vectors of training data are used to train a final decision classifier in the same way as it is used in the last section. We train 15 pairs of GMMs as the back-end classifiers.

Given an unknown speech utterance $O$, the confidence of a test utterance for target language $i$ is given by the likelihood ratio

$$
\lambda_{DUR,i} = \log\left(\frac{p(O|m^+, i)}{p(O|m^-, i)}\right)
$$

(4.16)
where \( p(O|m^+, i) \) is the likelihood score for the back-end GMM of target language \( i \), and \( p(O|m^-, i) \) is the likelihood score for the corresponding negative model. The unknown utterance \( O \) is classified to the language \( k \) which have the highest language confidence

\[
\lambda_{DUR,k} = \max_i (\lambda_{DUR,i})
\]  

(4.17)

### 4.3.5 Pitch

Pitch feature is another important prosodic feature. It has been used in some speaker recognition tasks [64], but has not successfully used in LID task yet. We initially design pitch features for Chinese dialect identification as Chinese dialects are largely differentiated by different intonation schemes. Inspired the promising results [65] in dialect Chinese dialect identification, we adopt this method in our LID system.

The conventional method to extract pitch feature is measure the energy and the dynamic of the energy. In front-end feature extraction, we calculate the autocorrelations of two adjacent frames and the covariance between them is estimated to extract the multi-dimensional tone features, we denote this extracted feature as pitch feature.

Given two vectors of speech data for two adjacent frames \( x_t(n) \) and \( s_{t+1}(n) \), \( n = 0, ..., N - 1 \), where \( N \) is the number of samples in the frame, \( t \) is the frame index. The process of pitch feature extraction consists of the following steps:

**Step 1** Calculate power density spectrum

\[
P_t(k) = |DFTx_t(n)|^2, k = 1, ..., K - 1
\]  

(4.18)

where \( K \) is the maximum number of DFT point we calculated.

**Step 2** Emphasis the low frequency by passing through a low-pass filter

\[
\tilde{P}_t(k) = P_t(k) \cdot W(k)
\]  

(4.19)

where \( W(k) = 1 + \cos(2\pi k/K) \), so that the dominant harmonics can be enhanced.

**Step 3** Normalize the power density spectrum

\[
\bar{P}_t(k) = \frac{\tilde{P}_t(k)}{\sum_{k=0}^{K-1} P_t(k)}
\]  

(4.20)
**Step 4** Calculate autocorrelation by applying inverse DFT

\[ R_t(k) = \text{DFT}^{-1} \bar{P}_t(k) \] (4.21)

**Step 5** Finally the pitch features are defined as follows

\[ c(d) = \frac{C}{K-d} \left[ \sum_k R_t(k) \cdot R_{t-1}(k+d) - \frac{1}{K-d} \sum_k R_t(k) \cdot \frac{1}{K-d} \sum_k R_{t-1}(k+d) \right] \] (4.22)

where \(-D \leq d \leq D\) is the index of the pitch feature, and the dimension of pitch feature vector is \(2D + 1\), in this system, we choose \(D = 5\), so the dimension of our pitch feature vector is 11. \(C\) is a constant to normalize the range of the pitch feature, chosen as \(10^8\).

The pitch features are modelled using the similar method as used in Duration and SDC features. For a given utterance, 11 dimensional pitch features are extracted from each frame [65]. A Gaussian mixture model (UBM) is trained using feature vectors from all languages. Then a GMM model is adapted from the UBM model for each target language. As a result, our back-end classifiers has 15 GMM models and one UBM model and all models have 16 Gaussian mixtures each.

An utterance is evaluated on the 15 GMMs to generate 15 language dependent scores in a 15-dimensional vector. The score vectors are normalized by the language independent scores which are generated by the UBM. The collection of training score vectors are used to train 15 pairs of final decision GMM classifiers.

Given an unknown speech utterance \(O\), the confidence of a test utterance for target language \(i\) is given by the likelihood ratio

\[ \lambda_{PIT,i} = \log(p(O|m^+, i)/p(O|m^-, i)) \] (4.23)

where \(p(O|m^+, i)\) is the likelihood score for the back-end GMM of target language \(i\), and \(p(O|m^-, i)\) is the likelihood score for the corresponding negative model. The unknown utterance \(O\) is classified to the language \(k\) which have the highest language confidence

\[ \lambda_{PIT,k} = \max_i(\lambda_{PIT,i}) \] (4.24)
4.4 Experiment Results

We conduct experiments on NIST 1996 and 2003 LRE data sets. In this section, we firstly compare the performance of the individual system, then we report the fusion result of these five features.

4.4.1 Individual System Performance

The experiment result on NIST LRE 1996 and 2003 data set of individual system described above are shown in table 4.1 and table 4.2.

The P-PRLM method gives best result as a single LID system, and it is also the most computational expensive method among five methods. The BOS method also captures phonotactic features, but in the different way as in P-PRLM. It requires single universal phoneme tokenizer, so the processing speed is faster than the P-PRLM system. Compared with the SDC feature based system, the BOS system achieves better accuracy for long testing utterances, but it’s not so efficient when identifying the short testing utterances. The SDC system captures low level acoustic information, it has steady performance on different length of the testing data. Compare with the SDC system, the performance of the P-PRLM and BOS system degrades greatly for the short testing utterances. Although the performance of the two prosodic features based system are worse than the other three systems, as both of them trained from small dimensional feature vectors, their have their advantage on the speed. And we also hope they compensate in the short test utterance when fuse them with the other three systems.

Table 4.1: EER of Individual system in 1996 LRE

<table>
<thead>
<tr>
<th>System/Duration</th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-PRLM</td>
<td>2.92</td>
<td>8.23</td>
<td>18.61</td>
</tr>
<tr>
<td>BOS</td>
<td>5.43</td>
<td>10.95</td>
<td>21.34</td>
</tr>
<tr>
<td>SDC</td>
<td>7.14</td>
<td>11.6</td>
<td>19.76</td>
</tr>
<tr>
<td>Duration</td>
<td>20.1</td>
<td>26.68</td>
<td>33.25</td>
</tr>
<tr>
<td>Pitch</td>
<td>35.89</td>
<td>39.14</td>
<td>39.78</td>
</tr>
</tbody>
</table>
Table 4.2: EER of Individual system in 2003 LRE

<table>
<thead>
<tr>
<th>System/Duration</th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-PRLM</td>
<td>4.54</td>
<td>11.31</td>
<td>20.37</td>
</tr>
<tr>
<td>BOS</td>
<td>7.15</td>
<td>13.72</td>
<td>24.45</td>
</tr>
<tr>
<td>SDC</td>
<td>8.47</td>
<td>13.1</td>
<td>21.19</td>
</tr>
<tr>
<td>Pitch</td>
<td>25.09</td>
<td>29.78</td>
<td>37.17</td>
</tr>
</tbody>
</table>

4.4.2 Analysis on P-PRLM and BOS approach

Although both P-PRLM and BOS captures the phonotactic information, they model the phonotactic in different ways. Specifically, they are different in terms of the front-end feature extraction and back-end classifiers. In this section, we will clarify the difference between the front-end and back-end of two approaches. The combinations of front-end and back-end are studied, corresponding experiment results are reported.

A front-end converts spoken utterances into sequences of token symbols. The P-PRLM system employs parallel phone recognition (PPR) front-end to derive multiple phonotactic features, we denote it as PPR front-end. It is followed by a set of n-gram phone language models (LM) that impose constraints on phone decoding and provide language score. The language models and the classifier are referred to as the back-end, here we denote it as LM back-end.

The BOS approach uses a universal phone recognition (UPR) to segment utterances into sequences, we denote it as UPR frontend. The sound sequences are converted into n-gram count vector, in this way, the LID problem becomes a vector-based classification problem. Many classifier designs exist in the machine learning literature for high-dimension vector classification. In this report we use support vector machine (SVM) as the backend classifier [31] [32], we denote it as SVM back-end.

Table 4.3: EER of front-end and back-end combination in 1996 LRE

<table>
<thead>
<tr>
<th>System/Duration</th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPR-SVM</td>
<td>2.75</td>
<td>8.23</td>
<td>21.16</td>
</tr>
<tr>
<td>PPR-LM</td>
<td>2.92</td>
<td>8.39</td>
<td>18.61</td>
</tr>
<tr>
<td>UPR-SVM</td>
<td>4.87</td>
<td>11.18</td>
<td>22.38</td>
</tr>
<tr>
<td>UPR-LM</td>
<td>6.78</td>
<td>15.9</td>
<td>27.20</td>
</tr>
</tbody>
</table>
Table 4.4: EER of front-end and back-end combination in 2003 LRE

<table>
<thead>
<tr>
<th>System/Duration</th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPR-SVM</td>
<td>3.62</td>
<td>10.36</td>
<td>21.25</td>
</tr>
<tr>
<td>PPR-LM</td>
<td>4.54</td>
<td>11.31</td>
<td>20.37</td>
</tr>
<tr>
<td>UPR-SVM</td>
<td>6.33</td>
<td>13.35</td>
<td>24.30</td>
</tr>
<tr>
<td>UPR-LM</td>
<td>10.24</td>
<td>19.23</td>
<td>30.28</td>
</tr>
</tbody>
</table>

Table 4.5: Speed of frontend and backend combination in 1996 LRE 30 seconds

<table>
<thead>
<tr>
<th>System</th>
<th>front-end</th>
<th>back-end</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPR-SVM</td>
<td>2.294</td>
<td>0.033</td>
</tr>
<tr>
<td>PPR-LM</td>
<td>2.394</td>
<td>0.091</td>
</tr>
<tr>
<td>UPR-SVM</td>
<td>0.983</td>
<td>0.005</td>
</tr>
<tr>
<td>UPR-LM</td>
<td>0.983</td>
<td>0.076</td>
</tr>
</tbody>
</table>

The individual system experiment result in Table 4.1 and 4.2 show that the BOS system has lower accuracy than P-PRLM system, but we can not claim that the PPR front-end and LM back-end of the P-PRLM system are better than UPR and SVM of the BOS system. It is interesting to compare the performance of the different combinations of these four front-end and back-end. Table 4.3 and 4.4 list the accuracy of these combination on NIST LRE 1996 and 2003 data set. Table 4.5 compares the time spent on the front-end and back-end for the 1996 30s test set.

Summarizing the results, we have the following observations:

- SVM back-end demonstrates advantage over LM back-end for the longer test utterances, while LM works better in shorter utterances. This result shows that the SVM favors longer utterances which provide richer long span phonotactic information.

- Although PPR front-end outperforms UPR front-end in general, UPR frontend shows advantage of computational efficiency in run-time operation over PPR frontend.

### 4.4.3 System Fusion

One of the solutions to fuse multiple features is the ensemble method. An ensemble of classifiers is a set of classifiers whose individual decisions are combined in the classification process. Our five-feature fusion LID system is formulated in this way.
To investigate how different levels of discriminative features complement each other, we use our P-PRLM classifier as the baseline, and then fuse other classifiers one by one into the ensemble. The fusion is carried by multiplying the likelihood ratio score from individual member classifiers. In the case of 5-feature fusion, for target language \( i \), we have

\[
\lambda_i = \alpha_1 \lambda_{P-PRLM,i} + \alpha_2 \lambda_{BOS,i} + \alpha_3 \lambda_{SDC,i} + \alpha_4 \lambda_{DUR,i} + \alpha_5 \lambda_{PIT,i}
\]  

(4.25)

where \( \alpha \) are the weighting parameters and \( \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 = 1 \). We use NIST 1996 LRE development data for fine-tuning the weighting parameters of the ensemble. The unknown utterance is classified to the language \( k \) which have the highest fusion score

\[
\lambda_k = \max_i (\lambda_i)
\]

(4.26)

where \( 1 \leq i \leq 15 \). Table 4.6 & 4.7 show the results for incremental fusion of ensemble with the last row being extracted from Singer et al [28] for comparison. Figure 4.3 shows the DET plots on 3-sec NIST 2003 LRE data. The proposed ensemble system significantly outperforms previous reported results on the 3-sec short test utterances and compare favorably on longer test utterances except 30-sec in 2003 LRE.

<table>
<thead>
<tr>
<th>System/Duration</th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-PRLM</td>
<td>2.92</td>
<td>8.23</td>
<td>18.61</td>
</tr>
<tr>
<td>P-PRLM + BOS</td>
<td>2.61</td>
<td>7.11</td>
<td>16.98</td>
</tr>
<tr>
<td>P-PRLM + BOS + SDC</td>
<td>2.38</td>
<td>6.8</td>
<td>15.70</td>
</tr>
<tr>
<td>P-PRLM + BOS + SDC + Duration</td>
<td>2.38</td>
<td>6.35</td>
<td>14.55</td>
</tr>
<tr>
<td>P-PRLM + BOS + SDC + Duration + Pitch</td>
<td>2.38</td>
<td>6.26</td>
<td>14.31</td>
</tr>
<tr>
<td>MIT fused system</td>
<td>2.7</td>
<td>6.9</td>
<td>17.4</td>
</tr>
</tbody>
</table>

Table 4.6: EER of system fusion on 1996 LRE

To look into the contribution of each member classifier in the ensemble, we can calculate the EER reductions by individual classifier when it is added into the ensemble, as shown in Table 4.8.

As both P-PRLM and BOS systems capture phonotactic features in different way, by fusing the two systems, we gain average 10.2% EER reduction evenly across the board. The SDC classifier captures low level acoustic information. The results also show that it also significantly contributes to EER reduction across the board. However, the effect is more obvious in 2003 LRE than in 1996 LRE. As for the prosodic based classifiers, we only see effect in 3-sec and 10-sec test cases.
Table 4.7: EER of system fusion on 2003 LRE

<table>
<thead>
<tr>
<th>System/Duration</th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-PRLM</td>
<td>4.54</td>
<td>11.31</td>
<td>20.37</td>
</tr>
<tr>
<td>P-PRLM + BOS</td>
<td>4.17</td>
<td>10.03</td>
<td>18.64</td>
</tr>
<tr>
<td>P-PRLM + BOS + SDC</td>
<td>3.27</td>
<td>8.55</td>
<td>16.66</td>
</tr>
<tr>
<td>P-PRLM + BOS + SDC + Duration</td>
<td>3.27</td>
<td>8.37</td>
<td>15.94</td>
</tr>
<tr>
<td>P-PRLM + BOS + SDC + Duration + Pitch</td>
<td>3.27</td>
<td>7.97</td>
<td>15.54</td>
</tr>
<tr>
<td>MIT fused system</td>
<td>2.8</td>
<td>7.8</td>
<td>20.30</td>
</tr>
</tbody>
</table>

4.4.4 Final Decision Classifiers

Each of the five member classifiers in the ensemble has a Gaussian mixture model (GMM) based classifier as the final decision classifier. For each target language we train two GMMs, the positive model and the negative model.

Table 4.9 illustrates the effects of the P-PRLM classifier on the number of mixtures in the positive and negative models. The leftmost column is the number of mixtures in the model pair, for instance, 32 − 128 refers to a pair of 32-mixture positive model and 128-mixture negative model. The experiment results show that increasing the mixture size
Table 4.8: EER reduction by member classifiers in the ensemble

<table>
<thead>
<tr>
<th>System/Duration</th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRE</td>
<td>1996</td>
<td>2003</td>
<td></td>
</tr>
<tr>
<td>P-PRLM</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BOS</td>
<td>10.6</td>
<td>8.1</td>
<td>13.6</td>
</tr>
<tr>
<td>SDC</td>
<td>8.8</td>
<td>4.4</td>
<td>14.7</td>
</tr>
<tr>
<td>Duration</td>
<td>0.0</td>
<td>0.0</td>
<td>6.6</td>
</tr>
<tr>
<td>Pitch</td>
<td>0.0</td>
<td>0.0</td>
<td>1.4</td>
</tr>
</tbody>
</table>

of the negative model consistently reduces EER. This can be explained by the fact that there are about 14 times more data in the negative model, which therefore needs more mixtures to capture the complexity. We have the same observations when constructing other classifiers in the ensemble.

Table 4.9: P-PRLM 30s EER on the number of mixtures for true model versus negative model

<table>
<thead>
<tr>
<th>Model</th>
<th>2003 30s</th>
<th>1996 30s</th>
</tr>
</thead>
<tbody>
<tr>
<td>32-32</td>
<td>5.70</td>
<td>3.90</td>
</tr>
<tr>
<td>32-128</td>
<td>5.26</td>
<td>3.58</td>
</tr>
<tr>
<td>32-512</td>
<td>5.00</td>
<td>3.42</td>
</tr>
<tr>
<td>64-128</td>
<td>5.14</td>
<td>3.54</td>
</tr>
<tr>
<td>64-512</td>
<td>4.54</td>
<td>2.92</td>
</tr>
<tr>
<td>128-512</td>
<td>4.7</td>
<td>3.1</td>
</tr>
</tbody>
</table>
Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this report, we analyzed the problem formulation of the LID and explored the discriminative features that human used to distinguish languages. In chapter 3 we studied the state-of-the-art LID systems and compared their performance on NIST 1996 and 2003 LRE data sets. After that, we presented our language identification system that fuses three levels discriminative information.

We have shown that different levels of information provide complementary language identification cues. It is found that P-PRLM and BOS features complement each other to fully explore both n-gram local phonotactics and utterance level collective phonotactic statistics. The spectral feature also consistently contributes to the LID tasks. It is found that fusing the lower level acoustic information and high level phonotactic information greatly improves the overall system. We have also successfully integrated the prosodic features into the LID task. The experiment results show that even the simple prosodic feature as pitch and phoneme duration are useful, especially for short speech segments.

The performance of proposed ensemble LID system on NIST 1996 and 2003 LRE datasets are comparable with the best system reported in the literature. The experiments in this paper also re-affirm, from a different angle, the findings in other reports [28] that spectral and phonotactic features are the most effective features for LID.
5.2 Future Works

Though the system developed in this report has proven to be competitive with other current LID systems, there are still many improvements that can be made. In particular, future research will attempt to satisfy the following goals:

(i) **Extend the findings to Speaker Identification** The first year of this study concentrates the methods in language identification. We believe the finding in LID will also benefit to the SID task. We will continue the current LID research, at the same time we will work on the SID task.

(ii) **Investigate feature extraction method** In the literature study, we found that adopting new feature extraction method significantly improve the performance of LID. We will continue to investigate new feature extraction method that specially useful in LID task.

(iii) **Extend the prosodic feature based component** Our experimental result shows that the prosodic information complement with the acoustic and phonotactic information in language identification. However, the prosodic based component still has low accuracy as single LID system. Since we are using very simple prosodic features, we can enhance the prosodic feature based system, for example, we can extract prosodic features in different angles, e.g., model the rhythm feature in the duration based system.

(iv) **System fusion** Currently, we use Gaussian Mixture model classifiers as the back-end classifiers for all our five components. Many other pattern classification classifier can also be used as the classifiers. Higher level of system fusion need to be explored. For example, fuse the individual system in the feature level.

5.3 Schedule of Future Research

The research plan for the next three years are shown in Figure 5.1. Our research will follow this plan, but some variation would be possible due to unexpected findings and difficulties.
### Chapter 5. Conclusions and Future Work

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>2006</th>
<th>2007</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Q1</td>
<td>Q2</td>
</tr>
<tr>
<td>1</td>
<td>Extend the findings to Speaker Identification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Subband feature extraction in LID and SID.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Study on better system fusion method; Apply the findings in LID to SID.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Thesis Writing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5.1: Schedule for future research**
Publication

Accepted

(i) Rong Tong, Bin Ma, Donglai Zhu, Haizhou Li and Eng Siong Chng “Integrating Acoustic, Prosodic and Phonotactic features for Spoken language identification”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006), May 14-19 Toulouse, France

(ii) Bin Ma, Donglai Zhu, Rong Tong “Tone Features Based on Pitch Flux for Chinese Dialect Identification”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006), May 14-19 Toulouse, France
References


REFERENCES


55
REFERENCES


REFERENCES

[29] T. J. Hazen and V.W. Zue. ”Recent improvements in an approach to segment-based automatic language identification”. ICSLP, Yokohama, Japan, 1994


57
REFERENCES


58
REFERENCES


[58] Linguistic Data Consortium (LDC), “The CallFriend corpra”, http://www.ldc.upenn.edu/Catalog/byType.jsp#speech.telephone

[59] Linguistic Data Consortium (LDC).http://www.ldc.upenn.edu/

[60] Language Identification Corpus of the Institute for Infocomm Research


REFERENCES


[67] Rong Tong, Bin Ma, Donglai Zhu, Haizhou Li and Eng Siong Chng “Integrating Acoustic, Prosodic and Phonotactic features for Spoken language identification”, *ICASSP*, May 14-19 Toulous, France, 2006


[69] Richard O. Duda, Peter E. Hart, David G. Stork, “*Pattern Classification*”

