Text-Independent Bilingual Speaker Verification System

Ma Bin

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
System Engineering and Engineering Management

Supervised by

Professor Helen Meng

June 2003

Abstract of thesis entitled:

Text-Independent Bilingual Speaker Verification System

Submitted by Ma Bin

for the degree of Master of Philosophy

at The Chinese University of Hong Kong in June 2003

Speaker verification systems verify whether given utterances are from claimed speakers. This problem is important because accurate speaker verification techniques can be applied to many security systems. Compared with biometric methods like fingerprint or face recognition, speaker verification systems do not require expensive specialized equipment and are more effective, especially for remote identity verification.

This thesis describes the development of a text-independent and language-independent speaker verification system. Two investigations are conducted. First, the baselines of text-dependent speaker verification systems are built based on the YOHO corpus. The baselines include the use of the left-to-right hidden Markov model (HMM) and the Gaussian mixture model (GMM). Left-to-right HMM is a statistical modeling technique, which is commonly used in text-dependent speaker verification. GMM is not concerned about the context information of what a speaker said in the speaker modeling process, and it has been proved to be an effective speaker model of text-independent speaker verification. In order to compensate for the variations of the

speaker's characteristics from trial to trial, we apply the technique of cohort normalization to the baseline systems. Second, this speaker verification system is bilingual, which is a novel feature but essential since Hong Kong is a multilingual city, where Cantonese and English are the two predominant languages. We have therefore developed a bilingual corpus - CUHK Bilingual Speech (CUBS) corpus - to investigate the language-dependency and bilingualism of the text-independent speaker verification system. The use of GMM modeling technique is expanded and applied to this task.

摘要

話者識別(Speaker Verification)系統可以通過聲音驗證說話人的身份。一個高識別率的話者識別系統在很多的安全系統中都有著潛在的應用價值。相對於生物測定學中的指紋識別，臉部識別來說，話者識別系統的優點在於不需要特殊的儀器，而且可以有效的適用於遠程的身份確認。

本論文紹了文本無關(text-independent)和語言無關(language-independent)的話者識別系統的建立過程。我們進行了兩項研究。首先，我們開發了基於 YOHO 語料庫的文本相關話者識別的基本系統。這些系統中分別採用了隱藏馬爾可夫模型(HMM)和高斯混合模型(GMM)。HMM 經常用於統計建模的文本相關話者識別系統中。GMM 在建模中不關注話者的說話內容。實驗證明，GMM 可以被有效的應用于文本無關的話者識別系統。說話人的語音特性在每一次試驗中都會變化，爲了減少這種影響，在基本系統中我們採用了最近同伴話者標準化技術。我們的話者識別系統的第二個新特徵是它的雙語性。香港是一個多語言地區，粵語和英語是兩種主要的語言。不受語言限制的話者識別系統會給使用者帶來很大便利。基於這一原因，我們開發了一個基於粵語和英語的雙語語料庫(CUBS)用於研究語言相關性，以及雙語文本無關系統的開發。在這個系統中，我們採用了 GMM 建模方法，並對其適用性進行了研究。

Acknowledgement

I would like to express a profound indebtedness to my supervisor, Professor Helen Meng, who provides numerous helpful guidance, precious ideas and hearty patronage to me. During these two years, she taught me not only how to do quality research, but also how to solve problems logically and efficiently. Moreover, the continuous encouragement and support from her also inspired me to overcome lots of difficulties and troubles. All of these contribute very much to the accomplishment of this research.

Also, I would like to express my deep gratitude to Dr. W.K. Lo for his valuable suggestions and technical assistance.

I do want to express a great appreciation to HCCL group members who work in cooperation and harmony with each other during these two years. Especially, I have to thank to my colleagues P.Y. Hui, O.Y. Mok, K. Xu and Y. Wang who have helped me in many different ways. Besides, I am also greatly indebted to other research helpers, such as C.K. Keung, M.C. Ho, T.Y. FUNG, T.H. Lo, K.F. Low, K.C. Siu, Ada Luk, Y.C. Li and Y.J. Li, for their kindly sharing knowledge and contribution to the database collection.

Finally, I would like to express a deep gratitude to my family and

friends for their continuous support and encouragement.

Contents

Abstract i

Acknowledgement iv

1 Introduction 1

1.1 Biometrics 2

1.2 Speaker Verification 3

1.3 Overview of Speaker Verification Systems 4

1.4 Text Dependency 4

1.4.1 Text-Dependent Speaker Verification 5

1.4.2 GMM-based Speaker Verification 6

1.5 Language Dependency 6

1.6 Normalization Techniques 7

1.7 Objectives of the Thesis 8

1.8 Thesis Organization 8

2 Background 10

2.1 Background Information 11

2.1.1 Speech Signal Acquisition 11

2.1.2 Speech Processing 11

2.1.3 Engineering Model of Speech Signal 13
2.1.4 Speaker Information in the Speech Signal 14
2.1.5 Feature Parameters 15
2.1.5.1 Mel-Frequency Cepstral Coefficients . 16
2.1.5.2 Linear Predictive Coding Derived Cepstral Coefficients 18
2.1.5.3 Energy Measures 20
2.1.5.4 Derivatives of Cepstral Coefficients . . 21
2.1.6 Evaluating Speaker Verification Systems 22
2.2 Common Techniques 24
2.2.1 Template Model Matching Methods 25
2.2.2 Statistical Model Methods 26
2.2.2.1 HMM Modeling Technique 27
2.2.2.2 GMM Modeling Techniques 30
2.2.2.3 Gaussian Mixture Model 31
2.2.2.4 The Advantages of GMM 32
2.2.3 Likelihood Scoring 32
2.2.4 General Approach to Decision Making 35
2.2.5 Cohort Normalization 35
2.2.5.1 Probability Score Normalization 36
2.2.5.2 Cohort Selection 37
2.3 Chapter Summary 38

3 Experimental Corpora **39**

3.1 The YOHO Corpus 39
3.1.1 Design of the YOHO Corpus 39
3.1.2 Data Collection Process of the YOHO Corpus . 40

3.1.3 Experimentation with the YOHO Corpus 41
3.2 CUHK Bilingual Speaker Verification Corpus 42
3.2.1 Design of the CUBS Corpus 42
3.2.2 Data Collection Process for the CUBS Corpus . 44
3.3 Chapter Summary 46

4 Text-Dependent Speaker Verification 47
4.1 Front-End Processing on the YOHO Corpus 48
4.2 Cohort Normalization Setup 50
4.3 HMM-based Speaker Verification Experiments 53
4.3.1 Subword HMM Models 53
4.3.2 Experimental Results 55
4.3.2.1 Comparison of Feature Representations 55
4.3.2.2 Effect of Cohort Normalization 58
4.4 Experiments on GMM-based Speaker Verification . . . 61
4.4.1 Experimental Setup 61
4.4.2 The number of Gaussian Mixture Components . 62
4.4.3 The Effect of Cohort Normalization 64
4.4.4 Comparison of HMM and GMM 65
4.5 Comparison with Previous Systems 67
4.6 Chapter Summary 70

5 Language- and Text-Independent Speaker Verification 71
5.1 Front-End Processing of the CUBS 72
5.2 Language- and Text-Independent Speaker Modeling . . 73
5.3 Cohort Normalization 74
5.4 Experimental Results and Analysis 75
5.4.1 Number of Gaussian Mixture Components . . . 78

5.4.2 The Cohort Normalization Effect 79
5.4.3 Language Dependency 80
5.4.4 Language-Independency 83
5.5 Chapter Summary 88

6 Conclusions and Future Work **90**
6.1 Summary 90
6.1.1 Feature Comparison 91
6.1.2 HMM Modeling 91
6.1.3 GMM Modeling 91
6.1.4 Cohort Normalization 92
6.1.5 Language Dependency 92
6.2 Future Work 93
6.2.1 Feature Parameters 93
6.2.2 Model Quality 93
6.2.2.1 Variance Flooring 93
6.2.2.2 Silence Detection 94
6.2.3 Conversational Speaker Verification 95

Bibliography **102**

List of Figures

1.1 Architecture of biometrics system. 3

1.2 Overview of a speaker verification system. 5

2.1 Pre-processing of speech signals. 12

2.2 An engineering model for speech production. (This figure is cited from [18].) 13

2.3 Illustration of an engineering model for speech production in the frequency domain. 14

2.4 Filter bank analysis. 16

2.5 Derivation of MFCC. 17

2.6 All-pole system for speech signal production. 19

2.7 Flowchart for calculating FAR and FRR. 22

2.8 FRR and FAR plot for speaker 104 in the YOHO corpus. 24

2.9 Left-to-right hidden Markov model. 28

2.10 Possible paths generated by an HMM in forced alignment. (This figure is cited from [50].) 34

3.1 An example of data collection interface for the CUBS corpus. 45

4.1 Comparison of two kinds of feature representation setups (LPCCs VS MFCCs). 49
4.2 Setup of cohort normalization. 51
4.3 Silence model and short pause model. 54
4.4 Feature parameters comparison of speaker verification on male and female speakers for the YOHO corpus. . . 56
4.5 Effect of cohort normalization for speaker verification with the YOHO corpus. 59
4.6 the effect of different numbers of Gaussian mixture components on TDSV. 63
4.7 Performance Comparisons of HMM- and GMM-based speaker verification. 66
4.8 Comparison of our speaker verification performance with other systems. 68

5.1 Cohort speakers selection. 75
5.2 The effects of mixture component numbers and cohort size in English. 77
5.3 The effect of mixture component numbers and cohort size in Cantonese. 78
5.4 Experimental results for testing language-dependency in English and Cantonese. 81
5.5 Enhanced verification performance by increasing the number of Gaussian mixture components M. 84
5.6 Evaluation of the language-independent TISV system. . 86

List of Tables

3.1 Example of recording data by means of answering question. 43

3.2 Example of recording data of Commands 43

3.3 Composition of enrollment data and verification data for each speaker. (The three versions of data include short, medium and long versions of answers and commands. 'E' refers to English. 'C' refers to Cantonese.) 44

4.1 List of subword HMM models in the YOHO database. . 54

4.2 Feature parameters comparison of speaker verification on male and female speakers for the YOHO corpus. . . 56

4.3 Results of applying cohort normalization for speaker verification with the YOHO corpus. 58

4.4 The effect of different numbers of Gaussian mixture components on TDSV. 63

4.5 The Effect of Cohort Normalization on GMM-based TDSV ($M = 64$). 65

4.6 Performance Comparisons of HMM- and GMM-based speaker verification. 66

4.7 Comparison of our speaker verification performance with other systems. 68

5.1 The effects of mixture component numbers and cohort size in English. 76
5.2 The effect of mixture component numbers and cohort size in Cantonese. 77
5.3 Experimental results for testing language dependency in English and Cantonese. 81
5.4 Comparison of M for language-independent TISV systems. 84
5.5 Evaluation of the language-independent TISV system. . 85

Chapter 1

Introduction

Speaker verification (SV) is a specialization of biometrics [41] [49]. Biometrics is the study of how human biological traits can be used to enhance security systems, such as verifying a person's identity by his/her voice. Reasonable use of biological traits can facilitate our life greatly. However, techniques based on such traits face many difficulties. For example, verifying identity through speech may be compromised due to differences in production behavior. Variations of speech production behavior may be due to gender and language differences, or outside effects, such as background noise and channel transmission effects. The aim of biometrics is to reduce variations in human biological traits. In this thesis, we will discuss this problem with regard to the scope of speaker verification. The Introduction thus outlines the study of biometrics in Section 1.1. Section 1.2 briefly introduces the concept of speaker verification. The overview of a speaker verification system is described in Section 1.3. Since the text-dependency and language-dependency of an SV system are the main subjects of investigation in this research, they will be introduced briefly in Section 1.4 and Sec-

tion 1.5. Finally, the objectives and overall thesis organization will be described in Section 1.7 and Section 1.8.

1.1 Biometrics

Biometrics is important in the field of person recognition because it provides the basis of categorizing distinctive personal characteristics, such as face, fingerprints, iris or voice. The advantage of a biometrical approach is that these characteristics cannot be forgotten, lost or stolen. Moreover, reasonable use of biometrics may make access and checking procedures faster and more reliable, thus substantially enhancing identity recognition accuracy.

In order to recognize an individual using biometrical characteristics, it is necessary that a recognition system has prior knowledge of an individual's biometrical characteristics. Hence, the development of biometric systems involves two separate modules: an enrollment module and a recognition module, which can be also called training and testing modules respectively (Figure 1.1).

Both the enrollment module and the recognition module include feature extraction sub-modules, which convert the biometrical characteristics into a set of biometrical features. With regard to the latter, better recognition performance is likely when a subject's features are more distinctive. However, before recognition is possible an enrollment module is responsible for enrolling new individuals into the system. The first stage of enrollment involves the individual supplying a number of samples of his/her biometrical characteristics as training data. A model of the individual is then built, based on the biometrical features

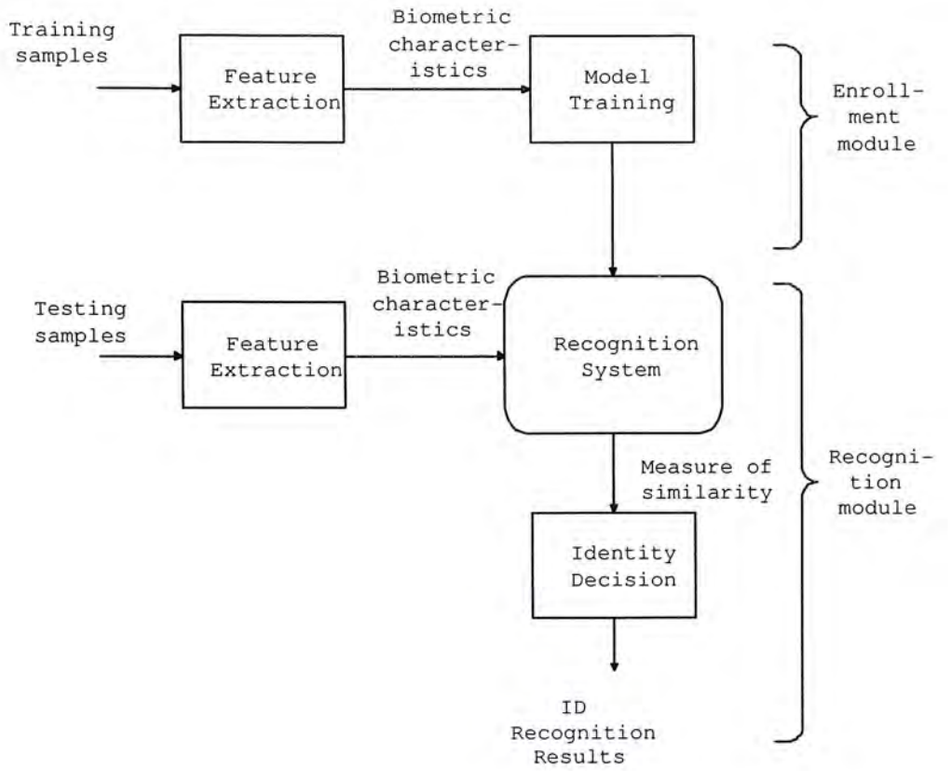

Figure 1.1: Architecture of biometrics system.

extracted from the supplied samples. At the stage of recognition, the individual first supplies a number of biometrical test samples. Then a similarity measure will be computed between the features of the test samples and the available models to estimate the identity of the individual.

1.2 Speaker Verification

Speech is a very rich medium of communication. Speech waves carry not only messages a speaker wants to express, but also the speaker's voice characteristics. From the point of view of biometrics, we can use a speaker's voice characteristics to recognize his/her identity by a speaker recognition system.

Speaker verification and speaker identification (SI) are the two main categories of speaker recognition. Speaker verification is the process of accepting/rejecting the identity claim of a speaker. Speaker identification is the process of determining from which of the registered speakers a given utterance comes. In this thesis, our investigation focuses on the speaker verification task.

1.3 Overview of Speaker Verification Systems

As a subset of biometrics, the architecture of a speaker verification system can be separated into two parts [39] - enrollment and recognition. The overview of a typical speaker verification system is shown in Figure 1.2. During enrollment, the reference template or model of a speaker is produced. For recognition, an identity claim is provided by an unknown speaker through an utterance. The speech waveform is then digitized to produce a sequence of vectors. Each of these vectors contains a set of acoustic features. A scoring algorithm then computes the similarity between the speech features of the unknown speaker and the reference template for the speaker whose identity is claimed. If the similarity score is above a certain threshold, the identity claim is accepted; otherwise, it is rejected. In brief, speaker verification involves three main techniques: feature extraction, speaker modeling, and a scoring technique.

1.4 Text Dependency

Speaker verification can be categorized into text-dependent and text-independent methods. The former requires a speaker to provide ut-

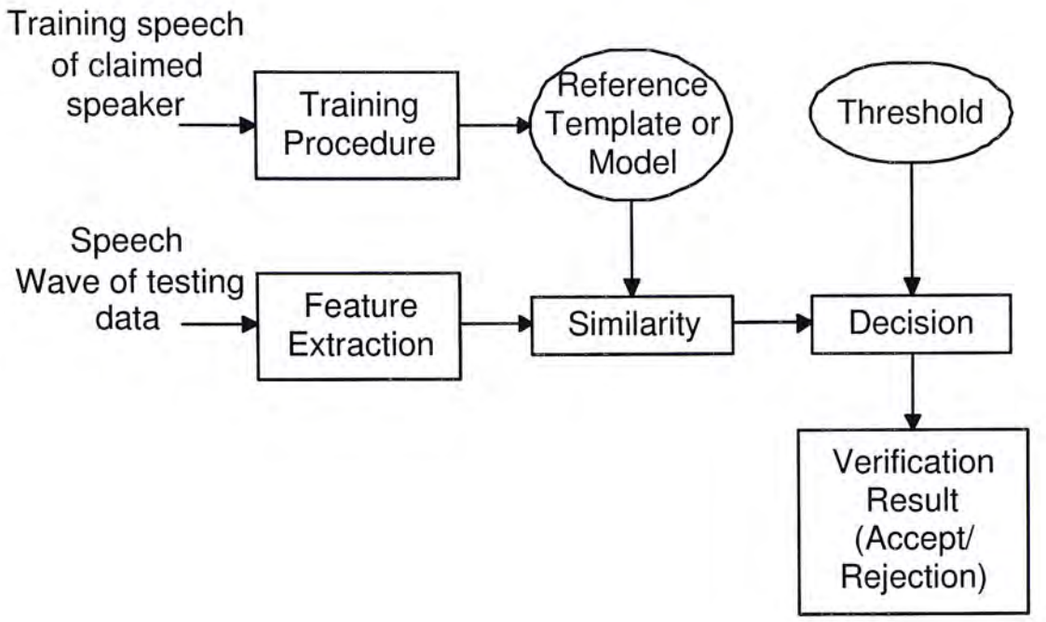

Figure 1.2: Overview of a speaker verification system.

terances of specific text for both enrollment and recognition, while the latter does not rely on any specific text being spoken.

1.4.1 Text-Dependent Speaker Verification

Text-dependent speaker verification (TDSV) can be categorized in two ways. The original TDSV uses predefined utterances for training and testing. This kind of SV system design is based on the assumption that a potential impostor is not able to record utterances from a legitimate client and then play them back. To prevent fraud via recording another kind of TDSV is often used; text-prompt speaker verification. This kind of TDSV is based on a fixed vocabulary such as digits. To avoid fraud, the text-prompt TDSV system prompts randomly generated digit strings and asks the client to repeat them (thus reducing the ability of an imposter to use recordings). To train the model of a speaker in the TDSV system, a set of words, subwords or phonetic Hid-

den Markov Models (HMMs) using Gaussian or multi-Gaussian distributions are typically employed [38]. This research also uses the HMMs as the major technique to develop a TDSV system. We use a corpus referred to as YOHO.

1.4.2 GMM-based Speaker Verification

Text-independent speaker verification (TISV) systems do not restrict users to any fixed or prompted phrases. Users have the freedom to say whatever they want in the verification process. The Gaussian Mixture Model (GMM) is a state-of-the-art modeling technique used in developing TISV systems. In a GMM-based SV system, the distribution of the feature vectors extracted from a person's speech is modeled with Gaussian mixture densities. Thus, the objective of the verification is to find whether the probability of a given utterance of the GMM is large enough to accept the speaker's identity claim. The GMM approach disregards phonetic information during model training and testing, because it uses only one model to describe the entire acoustic space. The GMM-based SV system is therefore also investigated in this thesis. We first apply GMM modeling techniques to a text-constrained SV scenario. Following this we present the results of our research on the validity of the GMM approach for TISV.

1.5 Language Dependency

Previous work on speaker verification research mostly focuses on a single language scenario, most commonly English. In Hong Kong, English and Cantonese are both commonly used. A speaker verification

system that is not constrained by language differences has significant value. For example, in a bilingual speaker recognition system, speakers can use their preferred language for the purposes of identification or verification trials. This technique will raise the usability of a verification system. Furthermore, some systems make use of user's private information to improve the security of the speaker verification system. This is referred to as verbal information verification (VIV) [19]. Language-independency is very valuable because it allows a VIV system to analyze a speaker's private information in different languages. A text-independent bilingual speech corpus (CUBS) is developed for research on the problem of language-dependency. This thesis investigates language-independent and text-independent speaker verification based on this corpus.

1.6 Normalization Techniques

Regardless of text-dependent or text-independent speaker verification, the most significant factor affecting verification performance is the characteristic variation of the speech signal from trial to trial. The variations may arise from the speakers themselves; for example, sickness, tiredness, emotional states and bad pronunciation, etc. Variations may also be due to differences in recording transmission, conditions, background noise, and so on. Moreover, speaker's voices change over the long term, adding additional variations. It is important for a speaker verification system to accommodate these variations. Through normalization, more reliable scores can be used in the decision making process. A detailed analysis of the normalization effect on the performance of

SV systems will be described in Chapter 4 and Chapter 5.

1.7 Objectives of the Thesis

A great deal of research has been conducted on speaker verification over many years [6] [39] [27]. However, most research on the topic focuses on a single language scenario. Hong Kong is a multilingual city, where English and Cantonese are both used extensively. Thus, bilingual speaker verification systems have substantial commercial potential. The goal of our study is to investigate speaker verification in the language-dependent and text-independent scenarios. In order to achieve this goal, a bilingual TISV system has been developed. Using this system, we can investigate the effect of language-dependency on the TISV system. Furthermore, bilingualism provides greater flexibility to current SV systems. We assume that both knowledge-based and behavior-based information can be used without language-limitation in an expanded verification system.

1.8 Thesis Organization

This thesis is organized as follows: Chapter 2 describes general knowledge about the common techniques that constitute an SV system. SV techniques encompassing speech signal processing, pattern recognition, hypothesis testing, and so on will also be reviewed. In Chapter 3, the specifications of the two experimental corpora, which are used in our SV system, will be described. A brief introduction to the text-dependent corpus - YOHO - will be reviewed first. The description of the development of our bilingual text-independent corpus - CUBS - follows.

Chapter 4 describes the development of two TDSV systems based on the YOHO corpus. The first is HMM-based, the second is GMM-based. A series of techniques in SV are compared, and the most suitable setup comprises our baseline TDSV system. Chapter 5 presents the extension of our TDSV system to a language-independent and text-independent SV system on CUBS corpus. We will demonstrate the scalability and validity of the GMM modeling technique for a bilingual TISV system. Conclusions and future work are presented in Chapter 6.

□ **End of chapter.**

Chapter 2

Background

Speaker verification (SV) is the process of accepting/rejecting the identity claim of a speaker. From the point of view of text dependency, speaker verification can also be categorized into text-dependent and text-independent tasks. This thesis sets out to develop a language-independent and text-independent speaker verification system. Generally speaking, there are five components in an SV system: digital speech data acquisition, feature extraction, enrollment to generate a speaker reference model, pattern matching, and a decision making process entailing acceptance or rejection [6]. In this chapter, we will describe basic information about and common techniques of speaker verification according to these areas. In Section 2.1, we will briefly introduce some general information about speaker verification systems. Front-end processing, the verification process, and the process of evaluating speaker verification systems will be introduced in this part. In Section 2.2, the commonly used speaker modeling and recognition techniques will be reviewed. The scoring and normalization techniques adopted in current SV systems are also described in this section.

2.1 Background Information

2.1.1 Speech Signal Acquisition

Sound traveling through air does so in the form of pressure waves. If we want to deal with sound using a computer, we need to transform an acoustic pressure wave into a digital signal. A microphone or telephone handset is often used to convert acoustic waves into an analog signal. This analog signal is conditioned with antialiasing filtering. According to sampling theory [12], antialiasing filters limit the bandwidth of the signal to approximately the Nyquist rate (half the sampling rate) before sampling. The conditioned analog signal is then sampled to form a digital signal by an analog-to-digital (A/D) converter. For example, telephone-quality speech with a sampling rate of 8 kHz can be represented by a spectrum with the maximum frequency of 4 kHz. Typically, for speech application, the sampling resolution varies from 8 to 16 bits at an 8kHz to 20kHz sampling rate [18].

2.1.2 Speech Processing

In order to be used by the recognition system, sampled waveforms must be converted into a sequence of feature vectors. Short time analysis of the speech signals [17] assumes that the speech signal is stationary within a short-time frame. In other words, in this frame, the speech signal has the same/similar acoustic properties. Hence, the feature vectors carrying the proper information can be extracted.

Generally speaking, the speech preprocessing includes three steps: signal pre-emphasis, frame blocking, and frame windowing (Figure 2.1).

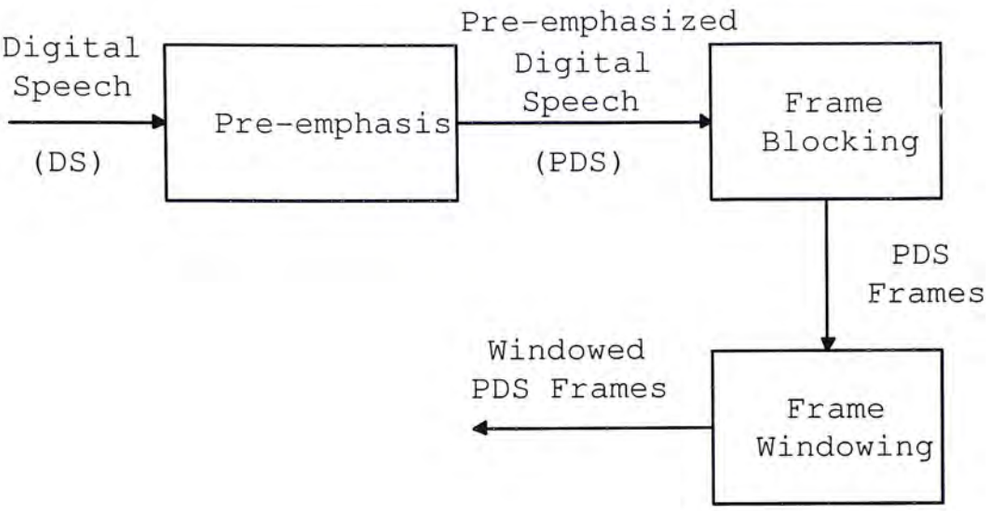

Figure 2.1: Pre-processing of speech signals.

Pre-emphasis processing of the speech signal employs a first order FIR filter. The pre-emphasized signal (S_{pe}) can be obtained using Eq.2.1:

$$S_{pe}(n) = s(n) - \alpha_{pe} s(n-1), \qquad \alpha_{pe} = 0.97 \tag{2.1}$$

The use of signal pre-emphasis is to improve the overall signal-to-noise ratio by minimizing adverse effects, such as attenuation differences or saturation of recording media, in subsequent parts of the system [12]. Previous work [22] has shown that pre-emphasizing a signal by a first order FIR filter can enhance the effectiveness of spectral features for a speaker recognition system.

Frame blocking is also called short-term processing. The pre-emphasized speech signal is separated into L successive overlapping frames by M samples Eq.2.2.

$$f(l,n) = S_{pe}(n + M(l-1)), \qquad n = 0, \ldots, N-1; l = l, \ldots, L \tag{2.2}$$

where the blocking window size is defined as N/F_s second, the frame rate is defined as M/F_s second, N is the number of samples, and F_s is

the sampling frequency. In order to minimize the signal discontinuities between neighboring frames, frame windowing will be used on blocked frames as shown in Eq.2.3.

$$f_w(l,n) = f(l,n)w(n), \qquad n = 0,\ldots,N-1 \tag{2.3}$$

A Hamming window $w(n)$[12] is often used to lower the effect emanating from the transitions between frames.

2.1.3 Engineering Model of Speech Signal

From an engineering point of view, speech production is a source-filter-radiation model, which is shown in Figure 2.2. Airflow, which is generated from the lungs, passing through the trachea and controlled by the opening and closing action of the vocal cords, is considered as the source of speech production. The source is an input of the vocal tract filter, which is modified by the positions of articulators to generate different sounds. Finally the sound is radiated by the lips.

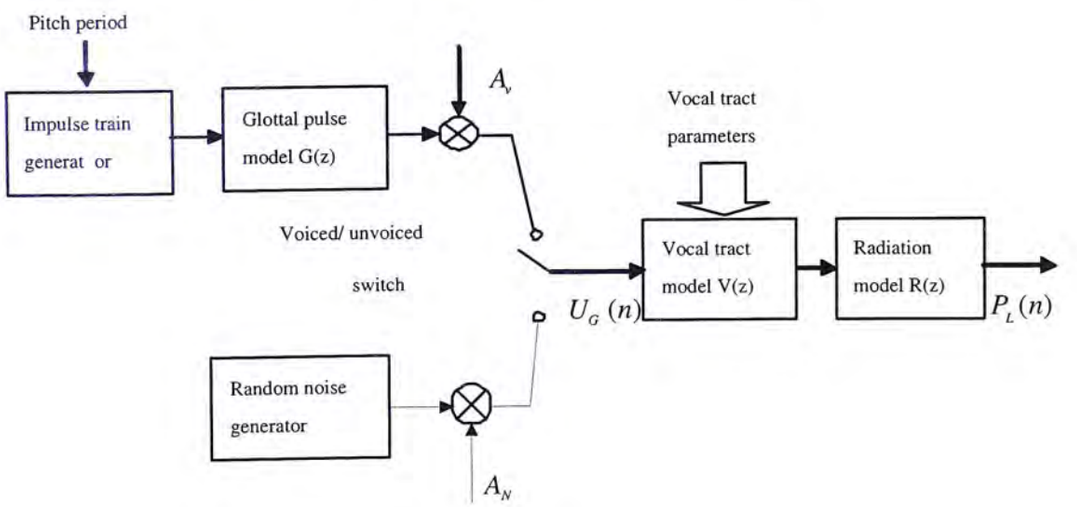

Figure 2.2: An engineering model for speech production. (This figure is cited from [18].)

The model is mathematically illustrated further in the frequency domain in Figure 2.3. In the time domain, the source is a string of im-

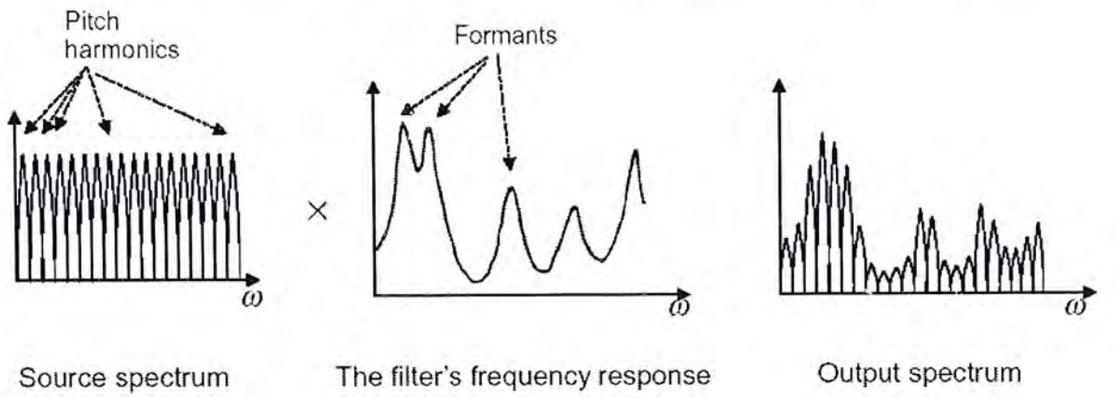

Figure 2.3: Illustration of an engineering model for speech production in the frequency domain.

pulses expressed as pitch harmonics in the frequency domain, which is an input signal to the vocal tract filter. Pitch is thus a characteristic of the excitation source. The vocal tract is modeled as a filter. Formants are the resonance frequencies of the vocal tract. They are viewed as peaks in the filter's frequency response. Hence, we use formants to describe the characteristics of the vocal tract filter. The final output spectrum is the multiplication of the source spectrum and the filter's frequency response.

2.1.4 Speaker Information in the Speech Signal

A speech signal carries not only the message we want to transmit, but also additional information such as emotion, language, identity of the speaker, and so on. This is why we can easily recognize a person by his/her voice.

Information in the speech signal that can be used to identify the

speaker is correlated with the physiological and behavioral characteristics of his/her speech production system. Humans always use characteristics such as psycho-linguistic peculiarities of the speaker for recognition. However, machines can only use physical characteristics of the speech production system to recognize a speaker's identity. In order to enable a machine to acquire the physical characteristics of speech production systems, such characteristics are usually represented in the form of specific feature parameters. Therefore, we need to convert speech signals into these feature parameters. We will discuss how to extract these feature parameters from speech signals in the sections that follow. Various types of feature parameters in common use will also be introduced.

2.1.5 Feature Parameters

As shown in the process of speech production, we deem that the information related to a speaker's identity is largely carried by the vocal tract and excitation characteristics [51]. We can use two typical speech analysis techniques to extract vocal tract characteristics; filter bank analysis and linear prediction analysis [25] [33]. The typical output of filter bank analysis is Mel-frequency cepstral coefficients (MFCC). The typical output of linear prediction analysis is linear predictive coding cepstral coefficients (LPCC) [11]. These two kinds of coefficient vectors can describe segmental information of the vocal tract. On one hand, the use of delta and delta-delta coefficients takes into account suprasegmental information, which is related with the short-term variation of segmental information [12]. On the other hand, a number of feature parameters such as energy and fundamental frequency are used

to describe a speaker's source information. These also perform well to discriminate speakers in speaker recognition systems, as has been proven in previous work [9][46].

In this section, we briefly present current knowledge about MFCC, LPCC, energy measurement, and their delta coefficients. These feature parameters are commonly used in speaker verification systems.

2.1.5.1 Mel-Frequency Cepstral Coefficients

MFCC can be generated using the filter bank analysis of speech signals. Filter bank analysis is a commonly used method for speech analysis. It aims to capture the spectral envelop by dividing the signal bandwidth into a number of bands and measuring the energy of each band. The process is shown in Figure 2.4

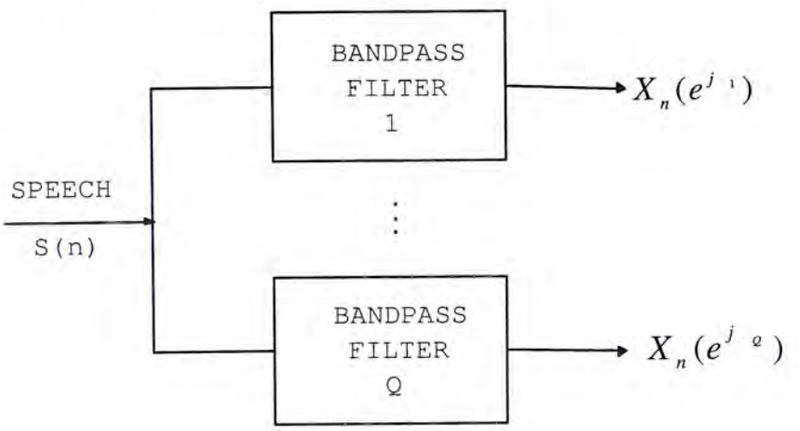

Figure 2.4: Filter bank analysis.

In speech perception, human beings are more sensitive to low frequency sounds. Ideally, in order to obtain more information about speech, the filter in low frequency should be narrow. Based on this idea, a popular filter bank coefficient named Mel-frequency Cepstral Coefficient [47] [48] is employed. Mel is the unit used to measure the

perceived frequency of a tone. Thus, in Mel-scale, the Mel-frequency is a non-linear map of the physical frequency as shown in Eq 2.4.

$$f_(Mel) = 2595log_{10}(1 + \frac{f_{Hz}}{700}) \tag{2.4}$$

The derivation of MFCC can be seen in Figure 2.5. Speech signals are short-time stationary. So speech processing is based on a short-time analysis basis; e.g., frames. The waveform is first processed by short-time windowing to obtain frame signals. The FFT computation is implemented to derive the spectrum of each frame's signal. The spectrum is analyzed by a set of Mel-scale spaced triangular-shaped filters (filter bank). The filter bank outputs correspond to the signal power in different frequency bands. Finally, such Mel-frequency coefficients are transformed into the cepstral domain where MFCC is derived.

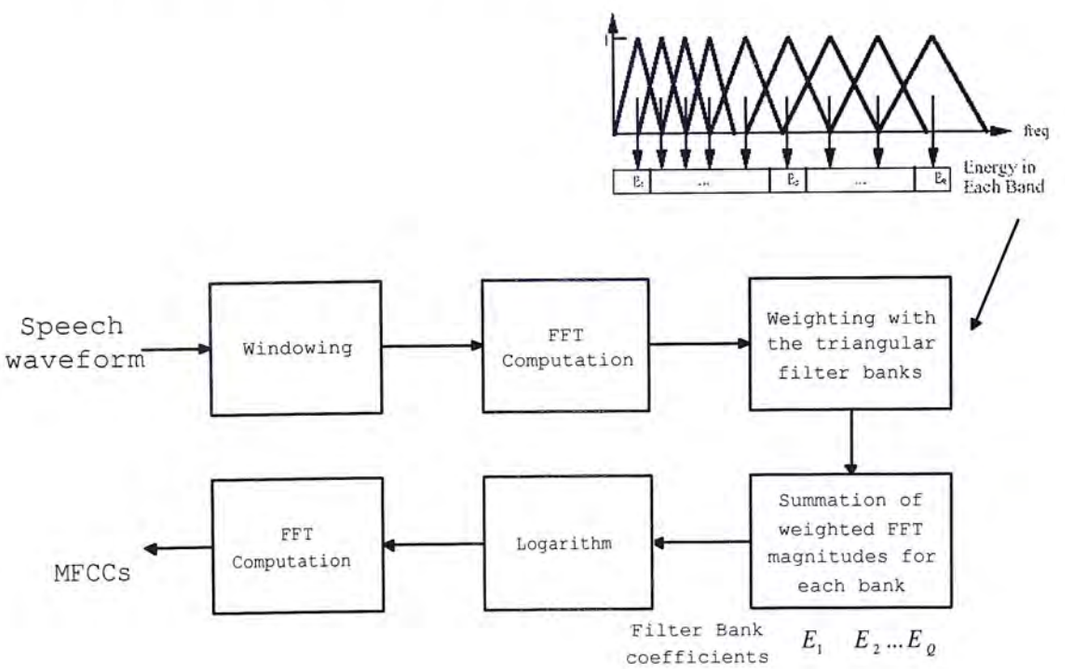

Figure 2.5: Derivation of MFCC.

Cepstral coefficients serve to separate vocal tract and source char-

acteristics from the speech spectrum. The speech spectrum is the multiplication of the excitation spectrum $E(\omega)$ and the vocal tract filter $V(\omega)$. The two components are linearly separated by a logarithm operation, from which we derive Eq.2.5.

$$\log S(\omega) = \log E(\omega) + \log V(\omega) \tag{2.5}$$

Cepstral coefficients are the Fourier transform representation of the logarithm spectrum. They are computed from filter bank outputs as Eq.2.6.

$$c_m = \frac{1}{I}\sum_{i=1}^{I} \log(E_i)[\frac{m\pi}{I}(i - \frac{1}{2})] \tag{2.6}$$

where I is the number of filters. E_i is the filter bank output of the ith filter, $1 \leq i \leq I$. c_m is the mth cepstral coefficient, $0 \leq m \leq M$, and M are the order of cepstrum (number of cepstral coefficients). Generally, I ranges from 20 to 40 and M from 8 to 16 in speech and speaker recognition.

By Fourier transform, we expect the envelop (slow-varied vocal tract) information and the detailed (quick-varied source) information to separate. It is understood that lower order cepstral coefficients describe the envelop, whereas the higher order ones are assumed to represent detail.

2.1.5.2 Linear Predictive Coding Derived Cepstral Coefficients

LPC derived cepstral coefficients have been used for short-time spectral measurement of speech processing for many years [36], and is derived from LPC theory [24] [28].

According to the LPC method, given speech samples at time n, $s(n)$ can be estimated as the linear combination of past p samples plus the

current excitation signal, $Gu(n)$, such that

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{2.7}$$

where $u(n)$ is a normalized excitation and G is the gain of the excitation. Expressing Eq.2.7 in Z-domain, we get Eq.2.8.

$$H(Z) = \frac{X(z)}{E(z)} = \frac{G}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}} \tag{2.8}$$

The vocal tract filter is described as an all-pole system as shown in Figure 2.6. p is the order of this system. $\{a_k\}$ is the coefficient to describe the digital vocal tract filter $H(z)$. Since a speaker's vocal tract changes slowly with time, $\{a_k\}$ are assumed to be constant over the speech analysis frames.

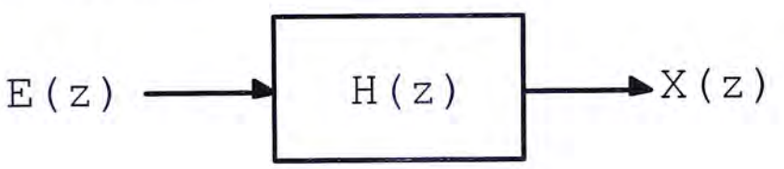

Figure 2.6: All-pole system for speech signal production.

To estimate $\{a_k\}$ in a case where $u(n)$ is unknown, we regard the predicted speech signal $\tilde{s}(n)$ as a linear combination of past p samples and thus defined as Eq.2.9.

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \tag{2.9}$$

The predicted error is $e(n)$,

$$e(n) = s(n) - \tilde{s}(n) = Gu(n) \tag{2.10}$$

which is equal to the scaled excitation.

LPC coefficients can be used to describe the vocal tract filter. However, it has been shown that if the order of LPC increases, the LPC

spectrum will attempt to accommodate individual pitch harmonics. This means that the order selection is important in LPC estimation [18]. LPCC is the cepstral coefficients of LPC and are computed recursively from LPC coefficients as Eq.2.11.

$$c_n = -\alpha_n + \sum_{i=1}^{n-1} \frac{(n-i)}{n} \alpha_i c_{n-i}, n \geq 1 \tag{2.11}$$

where $\alpha_i = 0$ when $i > k$.

Towards MFCC, cepstral coefficients are mainly used to separate the vocal tract component and the excitation component. Nevertheless, the LPC spectrum contains less excitation information. Therefore, using LPCC does not separate the vocal tract component from the excitation components, but is used to smooth the formant peaks in the LPC spectrum [9].

2.1.5.3 Energy Measures

To augment the spectral parameters derived from the Mel-filterbank or the linear prediction analysis, an energy term can be appended. Energy is computed as the logarithm of the signal energy [50]; that is, for speech samples $\{s_n, n = 1, \ldots, N\}$,

$$E = \log \sum_{n=1}^{N} s_n^2 \tag{2.12}$$

This log-energy measure can be normalized to the range -E_{min} to 1.0 by subtracting the maximum value of E in the utterance and adding 1.0.

2.1.5.4 Derivatives of Cepstral Coefficients

Cepstral representations only provide good approximations to the local spectral properties; i.e., the properties of the current frame. However, it has been shown that transitional spectrum information between frames is relatively complementary to instantaneous spectral information. Moreover, it is less affected by channel effects [17], and thus useful for speaker verification. Generally, we use derivatives of cepstral coefficients to describe the dynamic movement of the spectrum [18] [14] [45]. Investigations into speech and speaker recognition show that the performance of recognition systems can be greatly enhanced by adding time derivatives to the basic static parameters.

First- and second-order coefficients are typically used for derivative coefficients. The derivatives of the time functions of cepstral coefficients are extracted at each frame period to represent spectral dynamics, and are respectively called the delta and delta-delta cepstral coefficients. The delta coefficients are computed using the following regression formula

$$d_n^t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_n^{t+\theta} - c_n^{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2} \tag{2.13}$$

where d_n^t is a delta coefficient at time t, which is computed in terms of the correspondent static coefficients $c_n^{t-\Theta}$ to $c_n^{t+\Theta}$. The value of Θ equals the number of frame windows. The same formula is applied to the delta coefficients to obtain delta-delta coefficients.

2.1.6 Evaluating Speaker Verification Systems

The performance of an SV system is usually estimated by two kinds of error measures: false acceptance rate (FAR) and false rejection rate (FRR) [6] [39]. False acceptance occurs when the system incorrectly accepts an impostor, and false rejection occurs when the system incorrectly rejects a true speaker. Hence, FAR corresponds to the probability of accepting a speaker when he/she is an impostor, while FRR corresponds to the probability of rejecting a speaker when he/she is the true speaker.

Equal error rate (EER), which is obtained when FAR equals FRR, is often employed as the criterion to describe the performance of laboratory-developed speaker verification systems. Obviously, the lower the EER, the better the performance of a system.

The means by which to calculate FAR and FRR in an experimental speaker verification system is shown in Figure 2.7.

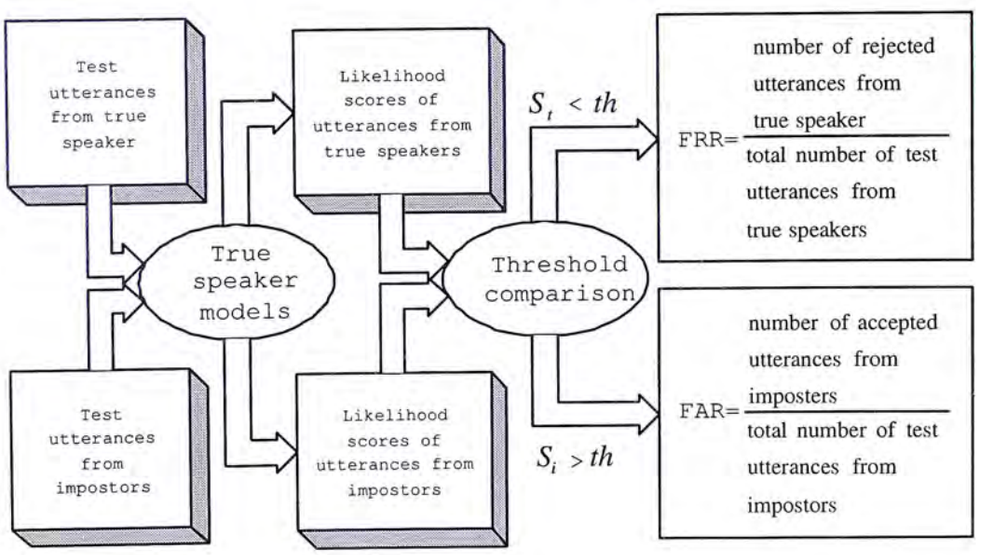

Figure 2.7: Flowchart for calculating FAR and FRR.

For experimental purposes, each speaker in the database is the true speaker in iteration. All the other speakers are treated as impostors. To calculate the FRR, all test utterances from the true speaker are scored against his/her own model to obtain the likelihood score for each utterance. The scores (S_t) are compared against a threshold (th). This threshold is an *a posteriori* threshold [13]; that is, it is set by determining the equal error rate threshold. The *a posteriori* threshold setting provides a way to evaluate the discrimination capabilities of a particular speaker model [8]. The scores smaller than the threshold ($S_t < th$) are marked. The number of this kind of score stands for the number of the falsely rejected trials of the true speaker. The FRR equals the percentage of falsely rejected trials in all trials (Eq.2.14).

$$FRR = \frac{\text{Number of rejected utterances from a true speaker}}{\text{Total number of test utterances from a true speaker}} \times 100\% \tag{2.14}$$

The tested utterances from all impostors are also scored against a true speaker's model. These scores (S_i) are compared with the threshold, after which FAR can be obtained by calculating the percentage of how many scores from among the total (S_i) are greater than the threshold ($S_i > th$) (2.15).

$$FAR = \frac{\text{Number of accepted utterances from impostors}}{\text{Total number of test utterances from impostors}} \times 100\% \tag{2.15}$$

Continuously adjusting the decision threshold and scoring with the comparison gives rise to FRR and FAR curves through tracking two kinds of error rate points. Finally, the EER value can be found at the intersection point of two curves. Figure 2.8 shows the FRR and FAR curves for one of the speakers in the YOHO corpus.

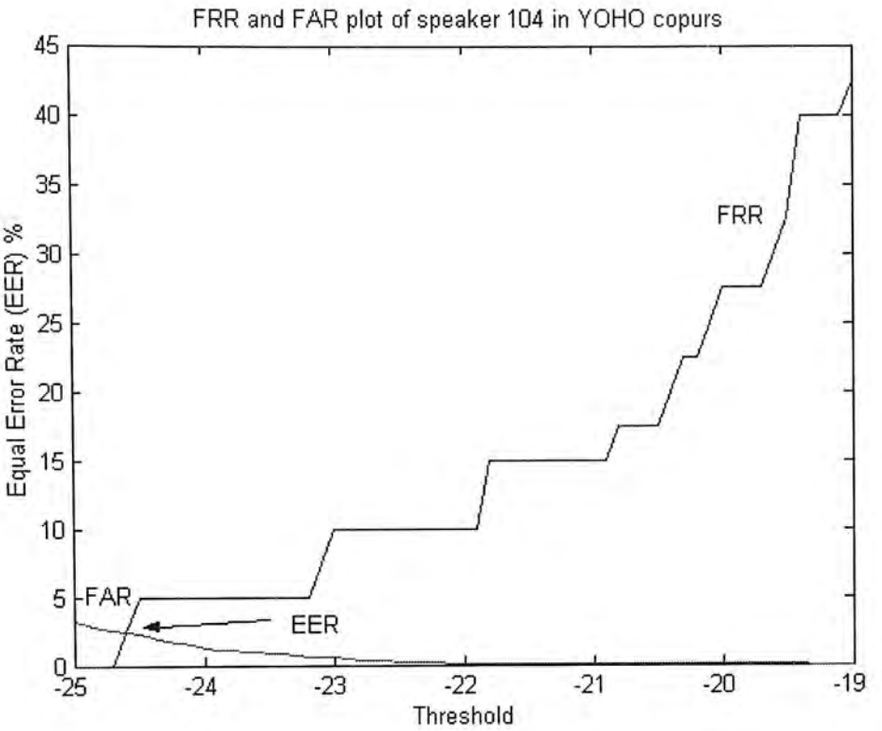

Figure 2.8: FRR and FAR plot for speaker 104 in the YOHO corpus.

This procedure is repeated N_{spk} times, where N_{spk} is the number of speakers in the database. When all the speakers are tested, the *mean EER* (Eq.2.16) of N_{spk} speakers will be computed to evaluate the performance of the speaker verification system.

$$Mean\ EER = \frac{1}{N_{spk}} \sum_{i=1}^{N_{spk}} EER_i \tag{2.16}$$

2.2 Common Techniques

Speaker verification systems range from small vocabulary text-dependent (TD) systems to large vocabulary text-independent (TI) systems. Despite differences in size, these recognition systems share a common problem that requires solving; that is, how to model a speaker using

the features extracted from his/her voice. In the main, two models have been used extensively in speaker verification systems: template models and statistical models. The template model attempts to model the speech production process in a non-parametric manner by retaining a number of feature vector sequences. Feature vectors are derived from multiple utterances by the same person of the same word. In the past, template modeling techniques have dominated efforts in developing TDSV systems because they are intuitively more logical. However, recent work has demonstrated that statistical models are more flexible and hence allow for better modeling performance. A statistical model regards the speech production process as a parametric random process and assumes that the parameters of the underlying stochastic process can be estimated in a precise, well-defined manner. In this section, several traditional and state-of-the-art techniques used in the speaker verification systems will be reviewed.

2.2.1 Template Model Matching Methods

Template model matching methods were dominant in the early research on TDSV works. Typical template model matching methods include dynamic time warping (DTW), vector quantization (VQ) source modeling, and the nearest neighboring method.

The DTW method is the most popular template-based system because of its ability to compensate for variable speaking rates [40]. In this approach, each utterance is represented by a sequence of feature vectors. The trail-to-trail variation of utterances of the same test is normalized by aligning the analyzed feature vector sequence of a test utterance to the template feature vector sequence using a DTW algo-

rithm. The overall distance between the test utterance and the template is the basis of decisions regarding recognition.

VQ source modeling uses multiple templates to represent a frame of speech [35] [30] [44]. In this approach, each speaker is represented by a code-book of spectral templates, which represent the phonetic sound clusters in his/her speech. The nearest neighbor method is always integrated into DTW and VQ-based methods [16].

2.2.2 Statistical Model Methods

Compared with template model matching methods, statistical model methods are more flexible. Theoretically, using the distribution of likelihood scores is more reasonable. Hence, this kind of method is widely employed in state-of-the-art speaker verification systems.

A very popular stochastic aproach for modeling the speech production process is the hidden Markov model (HMM). HMMs are an extension to conventional Markov models [1]. In the case of an HMM, the probabilistic function of an observed state is a doubly embedded stochastic process. The underlying stochastic process is not directly observable, which is why they are referred to as hidden [51]. Researchers have found left-to-right HMMs particularly useful for analyzing speech signals. One of the properties of left-to-right model is that as time increases, the state index increases or stays the same, which is why systems incorporating states proceed from left to right. Since the properties of a speech signal change over time in a successive manner, this model is very useful for modeling the speech production process. Hence, it is widely used in speaker verification systems.

Another popular approach is the Gaussian mixture model (GMM).

We can think of a GMM as a single state HMM. It provides a probabilistic model for the underlying sounds of a person's voice. But unlike HMMs, it does not impose any Markovian constraints between difference classes of sound. For instance, it has the capacity to discard phonetic information embodied in a speaker's utterances and use only specific characteristics. For this reason, GMM modeling techniques can be widely used in TISV systems.

After utilizing the stochastic approach (either HMMs or GMMs) to model speakers, the likelihood score of an observation is computed by the appropriate model. An observation here is a random vector, whose conditional probability density function depends on the speaker. Given the density function, the probability that an utterance is produced by a speaker can be accurately determined. In this section, we briefly introduce how to use HMMs and GMMs to model a speaker for a speaker verification system.

2.2.2.1 HMM Modeling Technique

The Hidden Markov model is a popular stochastic model for modeling acoustic speech units [38] [33] [32]. The main reason for this is that speech is behavior and thus exhibits a substantial degree of intra-speaker variance. For example, identical sentences uttered by the same speaker at different times may result in a similar yet different sequence of feature vectors. Hence, in order to cope with intra-speaker variation in a feature space and provide a robust representation of the speaker's characteristics, we use HMMs to model those characteristics by describing the speech production as a stochastic process. A brief description of HMMs and model training is provided in the following paragraphs.

HMMs consist of underlying Markov chains. The Markov chain is a finite set of states. The transitions between states are modeled by a transition probability matrix A, assuming that the probability of being in state s_i at time t depends only on the state occupied at time $t-1$. If the state probability vector $\vec{\pi}$ is known for $t = 0$, the probability vector for the next observation can be computed as

$$\vec{\pi_t} = A \cdot \vec{\pi_{t-1}} \Rightarrow \vec{\pi_t} = A^t \cdot \vec{\pi_0} \tag{2.17}$$

.

If any backward transition in the state sequences is not allowed (upper-triangular transition matrix), the model is referred to as a left-to-right Markov model, which is shown in Figure 2.9.

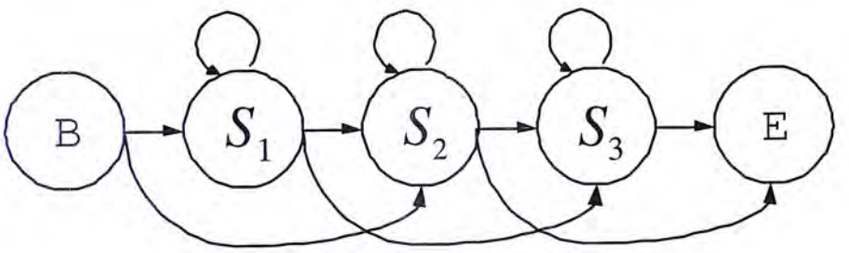

Figure 2.9: Left-to-right hidden Markov model.

The left-to-right HMM modeling technique is mostly applied in TDSV systems. An HMM differs from a Markov model in the sense that the state sequence cannot be observed directly (it is hidden) and only the observation (a feature vector representing certain speech waveform patterns) sequence is known. In HMMs, a probability density function (pdf) describes the probability p_i for observation $\vec{x}$ given that the process is in state s_i. p_i can be approximated by weighted sums of distributions:

$$p_i(\vec{x}) = \sum_{j=1}^{M} \omega_{ij} p_{ij}(\vec{x}), \qquad \text{where, } \sum_{j=1}^{M} \omega_{ij} = 1, \forall i \in [1, N] \tag{2.18}$$

where i is the state index, j is the mixture index, N is the number of states, and M is the number of mixtures. In most systems, p_{ij} has a Gaussian distribution:

$$p_{ij}(\vec{x}) = \frac{1}{\sqrt{(2\pi)^n |\sum_{ij}|}} \exp\{-\frac{1}{2}(\vec{x} - \vec{\mu_{ij}})' \sum_{ij}^{-1} (\vec{x} - \vec{\mu_{ij}})\} \tag{2.19}$$

where n is the feature dimensionality. The model parameters are denoted as w_{ij}, $\vec{\mu_{ij}}$, $\sum_{ij}$, A, where ω is the mixture weights, $\vec{\mu_{ij}}$ is the mean vector, $\sum_{ij}$ is the covariance matrix, and A the transition matrix.

The process of speaker model training is defined as the determination of the optimal model parameters given a set of training vectors from a particular speaker. Two approaches exist for training the HMM: the Baum-Welch algorithm [42] and the Viterbi algorithm - both of which use Maximum Likelihood (ML) criterion. Simply speaking, the training of the HMM involves assuming an initial estimation of the model λ. The assumption assigns values to the elements of $\vec{\pi}$ and matrix A. We can then re-estimate the model using known training samples. For each training sequence O, the parameters of the new model λ' are re-estimated from those of the old λ, until λ' is superior to the original model λ according to the training sequence. At each iteration, λ is replaced by λ' and another re-estimation takes place, until the parameters of the model are convergent.

Originally, an HMM was used extensively to model the fundamental speech units in speech recognition in order to adequately characterize

the varying nature of speech signals. However, in a speaker verification system, each speaker is typically represented by a set of HMMs. These HMMs are constructed by a single HMM that captures statistical properties of a number of speech components, such as phonemes, subwords, words, and so on. These HMMs are speaker-dependent models whose estimates are based on the speech data from enrollment session of the corresponding speaker. Hence, the characteristics of speech data for each speaker are then contained in speaker-dependent models. To verify a speaker's claim, the testing utterance is scored using the well trained speaker-dependent HMM. The score represents the probability that the observation sequence is uttered by a particular speaker, given the speaker-dependent HMM. Acceptance or rejection is then based on that score.

2.2.2.2 GMM Modeling Techniques

An investigation by Matsui and Furui in 1992 [26] showed that speaker recognition rates are strongly correlated with the total number of mixtures, irrespective of the number of states. This means that the information on transitions between different states is ineffective for text-independent speaker recognition. They investigated the case of using a signal-state multi-mixture Gaussian mixture model to characterize the speaker-specific features regardless of phonetic information [34]. In this work, we use a Gaussian mixture model technique to transform our HMM-based speaker verification. In this section, we begin by briefly describing the GMM. Second, we describe the advantages of using the GMM to model speakers in speaker verification systems.

2.2.2.3 Gaussian Mixture Model

The GMM is a category of density model that comprises a number of component functions (Gaussian) [34]. These component functions are combined to provide multi-model density. This property enables us to utilize the GMM method in text-independent speaker recognition systems.

A Gaussian mixture density for a feature vector $\vec{x}$, given the parameter vector λ, is a weighted sum of M densities. It is given by the equation:

$$p(\vec{x}|\lambda) = \sum_{m=1}^{M} p_m b_m(\vec{x}) \tag{2.20}$$

where $\vec{x}$ is a D-dimensional random vector, $b_m(\vec{x})$, $m = 1, \ldots, M$ is the component density, and p_m, $m = 1, \ldots, M$, is the mixture weights. Each component density is a Gaussian function of the form:

$$b_m(x) = \frac{1}{\sqrt{(2\pi)^{D/2}|\sum_m|^{1/2}}} \exp\{-\frac{1}{2}(\vec{x} - \vec{\mu_{mj}})' \sum_{mj}^{-1} (\vec{x} - \vec{\mu_{mj}})\} \tag{2.21}$$

with mean vector $\vec{\mu_m}$, and covariance matrix $\sum_m$. The mixture weights satisfy the constraint that $\sum_{m=1}^{M} p_m = 1$. The complete Gaussian mixture density is parameterized by the mean vector, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation:

$$\lambda = \{p_m, \vec{\mu_m}, \sum_m\} \tag{2.22}$$

In GMM speaker recognition systems, the distribution of feature vectors extracted from a person's speech is modeled as a Gaussian

mixture density. Thus, for speakers, who are represented by GMMs, the objective of recognition is to compare the posteriori probability of the model given the observation sequence. Compared with a traditional HMM approach, GMMs can also be treated as a signal state HMM with many mixture components. Hence, GMM systems use only one large model and allow the sharing of training data between different mixtures, disregarding phonetic-specific information. This leads to better-trained mixture parameters.

2.2.2.4 The Advantages of GMM

There are two principal advantages for applying Gaussian mixture densities as a representation of speaker identity [34]. The first is the intuitive notion that the individual component densities of a multi-model density may model an underlying set of acoustic classes (e.g., stops, fricatives, vowels, semivowels, nasals, etc.). These acoustic classes reflect general speaker-dependent vocal tract configurations that are useful for characterizing speaker identity. The second advantage is the empirical observation that a linear combination of Gaussian basis functions is capable of representing a large class of sample distributions. Hence, one of the powerful attributes of GMM is an ability to form smooth approximations to arbitrarily shaped densities. These advantages provide a good foundation for TISV.

2.2.3 Likelihood Scoring

For speaker verification systems, scores are used to verify the speakers according to the likelihood probabilities along the path of speech segments. More specifically, the score reflects the probability of observing

feature vectors across the path of segments in forced alignment from the transcription to a given utterance.

The concept of forced alignment mostly occurs in speech recognition [50]. For speech recognition, in order to train an acoustic model, we need a training set of labeled examples to use in classifying our acoustic models into a set of units (such as phonemes or other subwords, and so on). Occasionally, for a limited data corpus, the transcription can be determined manually. Following this, the transcripts of utterances are used to constrain an optimal alignment between existing speech models and new speech data. This process is called forced alignment.

The process of forced alignment [50] can be regarded as a constrained search, which assigns labels to the segments of an utterance. The labels make up the transcription. The search is constrained because the transcription is known *a priori* in a TDSV system. In other words, the system knows the content of speech in advance. The HMMs labeled by utterance will be used to calculate the probability of likelihood. During the search, the utterance score of each alignment is accumulated by the acoustic model's likelihood scores. These likelihood scores reflect the probabilities of observing feature vectors across the segments in the known alignment of labels. The path of the segments that corresponds to the highest likelihood score is chosen as the forced alignment of the correspondent utterance.

The idea of finding the optimal path using the highest score in forced alignment is employed in speaker verification systems, as shown in Figure 2.10, where the vertical dimension represents the states of an HMM and the horizontal dimension represents certain feature vector sequences in time and is converted from a segment of speech.

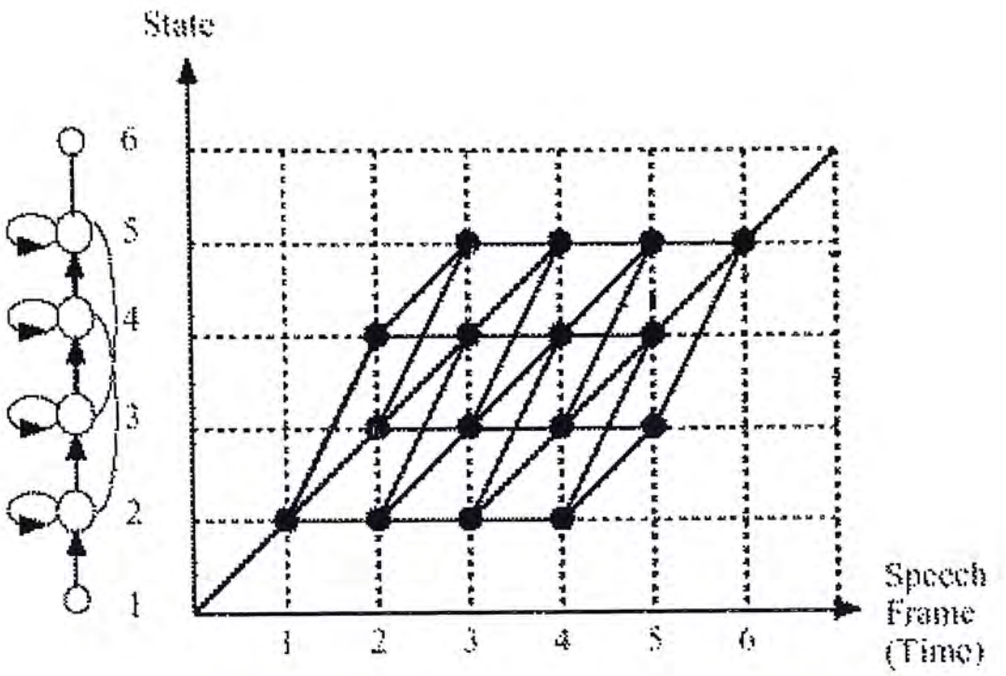

Figure 2.10: Possible paths generated by an HMM in forced alignment. (This figure is cited from [50].)

Each large dot in the figure represents the likelihood of observing that frame's feature vector at that time, and each arc between dots corresponds to a transition probability. The probability of any path is computed simply by summing the transition probabilities and the output probabilities along that path. The paths develop from left-to-right, column-by-column. There are many possible paths for the HMM to derive the feature vector sequence output. In speaker verification systems, the path accumulating the highest likelihood is based on the likelihood of the correspondent making that decision. Generally, a Viterbi algorithm is employed to implement the forced alignment process.

2.2.4 General Approach to Decision Making

Since speaker verification is a binary decision process, the conventional decision to accept/reject the claimed identity given an utterance can be made by comparing the log-likelihood score with a threshold (Eq. 2.23).

$$\log p(O|\lambda) \begin{cases} > \theta & acceptance \\ < \theta & rejection \end{cases} \tag{2.23}$$

where $p(O|\lambda)$ is the likelihood of the utterance O given the claimed speaker with model λ, and θ is the threshold for that speaker. If the score is larger than the threshold, the claim will be accepted, otherwise it will be rejected.

2.2.5 Cohort Normalization

With regard to the speaker verification tasks, the absolute likelihood score of an utterance from a speaker model is affected by many factors, such as a speakers' vocal characteristics, linguistic content, and speech quality. These factors make it very difficult to set a decision threshold for the absolute likelihood values that can be used over different verification tests. The likelihood ratio normalization produces a relative score. The score is more concerned with the utterance of a speaker and less volatile to non-speaker utterance variations. Hence, we expect that the distinctions between speakers will be more obvious and the threshold will be set more easily by normalization. The basic idea of cohort normalization will be described in the following paragraphs.

2.2.5.1 Probability Score Normalization

Variations in speaking behavior, recording, and transmission conditions may cause large variations in scores and make the assignment of appropriate thresholds even more difficult. Hence, a more reliable scoring method is proposed. Our approach is to apply a likelihood ratio test to an input utterance to determine if the claimed speaker is accepted or rejected. For an utterance O and a claimed speaker identity with correspondent model λ_C, the likelihood ratio can be defined as

$$\frac{Pr(O \text{ is from the claimed speaker})}{Pr(O \text{ is not from the claimed speaker})} = \frac{Pr(\lambda_C|O)}{Pr(\lambda_{\bar{C}}|O)} \tag{2.24}$$

Applying Bayes' rule and discarding the constant prior probabilities for claimed speakers and impostors, the likelihood ratio in the logarithm domain becomes

$$\Lambda(O) = \log p(O|\lambda_C) - \log p(O|\lambda_{\bar{C}}) \tag{2.25}$$

The term $p(O|\lambda_c)$ is the likelihood that the utterance is indeed from the claimed speaker. $p(O|\lambda_{\bar{C}})$ is the likelihood that the utterance is not from the claimed speaker. The likelihood ratio is compared to a threshold θ and the claimed speaker is accepted if $\Lambda(O) > \theta$, and rejected if $\Lambda(O) < \theta$. The likelihood ratio measures how much better the claimed speaker's model score for the tested utterance is compared to that of a non-claimed speaker's models.

Theoretically, the likelihood values between the input utterance and the models of a large number of speakers must be calculated. Unfortunately, the cost of computation becomes enormous when the number of reference speakers is large. Hence, there is no "anti-speaker" model [21], $\lambda_{\bar{C}}$, available. In order to reduce the amount of calculation, our

system adopts the cohort normalization technique, in which $\lambda_{\bar{C}}$ is estimated from a group of speakers. The models for this group of speakers are determined to be closest to or most "competitive" with the model of the claimed speaker. This group of speakers is referred to as cohort speakers, and are selected from the speakers in the database for each speaker. Thus, the likelihood of the utterance, given it is not from the claimed speaker, is formed using the collection of cohort speaker models. With a set of K cohort speaker models, $\{\lambda_1, \ldots, \lambda_K\}$, the log-likelihood of the normalization term is computed as

$$\frac{1}{K}\sum_{k=1}^{K}\log p(X|\lambda_k) \tag{2.26}$$

and thus Eq. 2.27 becomes

$$\Lambda(O) = \log p(O|\lambda_C) - \frac{1}{K}\sum_{k=1}^{K}\log p(X|\lambda_k) \tag{2.27}$$

.

2.2.5.2 Cohort Selection

In order to adopt a cohort normalization technique, it is necessary to solve the issue of how to select cohort speakers for each claimed speaker. A similarity measure is required to find the closest or most competitive speakers to the claimed speaker. A symmetrical distance (Eq.2.28) to measure the similarity between speakers' models is commonly adopted [34].

$$d(\lambda_a, \lambda_b) = \log\frac{p(O_a|\lambda_a)}{p(O_a|\lambda_b)} + \log\frac{p(O_b|\lambda_b)}{p(O_b|\lambda_a)} \tag{2.28}$$

Training utterances from two speakers can be used to determine similarity. For speaker a and b with model (λ_a, λ_b) and training utterance (O_a, O_b), we can use $d(\lambda_a, \lambda_b)$ to measure the divergence between them.

In Eq.2.28, the ratio $\frac{p(O_a|\lambda_a)}{p(O_a|\lambda_b)}$ measures how well speaker b's model scores with speaker a's speech relative to how well speaker a's model scores with his/her own speech. The more similar the models are, the smaller the ratio. The distance measurement then symmetrically combines the ratios comparing speaker a's and b's models. Hence, the selection of cohort speakers for each claimed speaker is based the pair-wise similarity between models.

2.3 Chapter Summary

In this chapter, we have provided background information about and some common techniques related to speaker verification systems. Two general approaches to front-end processing for speaker verification were reviewed. We have also presented popular approaches on modeling speakers in speaker verification systems. Statistical modeling techniques were reviewed in detail. Finally, the cohort normalization techniques in speaker verification system were described.

□ **End of chapter.**

Chapter 3

Experimental Corpora

In this chapter, two experimental corpora, YOHO and CUBS, are described. First, the introduction and experimentation of the YOHO corpus on text-dependent speaker verification are discussed. Following this, we describe the CUBS corpus, created by the author for making text-independent speaker verification experiments.

3.1 The YOHO Corpus

The YOHO corpus is designed for testing a prototype speaker verification system proposed with limited vocabulary [5]. It is distributed by the Linguistic Data Consortium (LDC). This Section presents a general introduction.

3.1.1 Design of the YOHO Corpus

There are 138 speakers (106 male speakers and 32 female speakers) in the YOHO corpus. They spanned a wide range of ages, job descriptions, and educational backgrounds. For each speaker, the collected

speech data was divided into enrollment and verification sets. Considering the long-term variation of speech characteristics, there were 4 enrollment sessions and 10 verification sessions involved. The period of collection for each speaker was three months, with a three-day interval between each verification session. Each enrollment session consisted of 24 utterances. Each verification session consisted of 4 utterances. Each speaker provided 96 enrollment utterances and 40 verification utterances.

The vocabulary employed in the YOHO corpus consists of two-digit numbers (doublets) in English (e.g. "thirty-four", "sixty-one", etc.). The doublets were chosen from a list that includes all the doublets from 21 to 99 with the following exceptions: (1) no exact decades (30, 40, etc.), (2) no double digits (22, 33, etc.), and (3) no numbers ending in "8" (28, 38, etc.).

The speech material of the YOHO corpus consists of "combination-lock" utterances for doublets; for example "35 - 72 - 41", pronounced "thirty five, seventy two, forty one". The average duration for such an utterance was approximately 2.5 seconds. There were optional pauses between doublets.

3.1.2 Data Collection Process of the YOHO Corpus

Speech recording for the YOHO corpus took place in the corner of a large room, in which low level noise could be heard [4]. The speech data of the YOHO database did not pass through a telephone channel. A high-quality simulated telephone recording system was used to perform data collection. All waveforms of the speech were low-pass filtered at 3.8 kHz and sampled at 8 kHz. Hence, the data for the

YOHO corpus is derived from simulating telephone quality speech in an office environment. Therefore, channel effects on the speech data were reduced.

3.1.3 Experimentation with the YOHO Corpus

The YOHO corpus is designed for evaluating text-dependent speaker verification. During the enrollment of the YOHO-based TDSV system, speaker models were developed using the speech data from enrollment sessions 1 to 4. In text-dependent scenarios, the enrollment data was used to train detailed acoustic models for each speaker. These acoustic models could be phoneme-based, subword-based, word-based, and so on. These constitute the speaker-specific models in the TDSV system. Subword models are often used for YOHO-based SV systems. For example, models for a given speaker's doublet in the YOHO corpus, "35", can be obtained without actually collecting the word model of "thirty-five". The subword model "thirty" and "five" can use the same subwords from other doublets, such as "thirty-nine" and "seventy-five". Therefore, the training data can be used more efficiently.

Another speaker modeling approach is to use GMMs to model the constrained vocabulary of the YOHO corpus. Hence, the entire acoustic space in the YOHO corpus can be modeled by only one GMM for any given speaker. The speaker's specific characteristics can be distinguished by this GMM.

When the speaker-specific model is developed, the testing process can be performed using the data from the verification sessions. A single testing trial can involve one utterance or all four utterances in the verification session. False-rejection measurement is calculated on

the basis of trials related to the speaker's own verification data. False-acceptance measurements are calculated on the basis of trials related to other speakers' verification data. According to the false-rejection and false-acceptance measurements, EER can be obtained and used to evaluate the performance of speaker verification systems.

3.2 CUHK Bilingual Speaker Verification Corpus

In order to build a bilingual TISV system, we developed a bilingual speaker verification corpus. We named this the CUHK Bilingual Speaker Verification (CUBS) corpus. A description of CUBS is presented below.

3.2.1 Design of the CUBS Corpus

The CUBS corpus consisted of 16 speakers (10 males and 6 females). Their ages and educational backgrounds were similar. Each speaker participated in 3 enrollment sessions for training and 1 verification session for testing. Each enrollment session consisted of 168 utterances. Hence, 504 utterances in total were used for the enrollment of each speaker. The verification session consisted of 78 utterances for each speaker.

We designed a scheme for the recording of the CUBS corpus, since there is no constraint to what speakers say in text-independent speaker verification. This consisted of designing several questions and tasks for recording. Data collection was performed by means of answering questions or giving commands to specific tasks. We set questions related to the speaker's private information, such as "favorite color" and "favorite food", etc. An example of the recording data by means of answering

question is shown in Table 3.1.

Question:	What is your favorite color?	你最喜欢什么颜色?
Short answer:	Purple.	紫色.
Medium answer:	It's purple.	我喜欢紫色.
Long answer:	My favorite color is purple.	我最喜欢的颜色是紫色.

Table 3.1: Example of recording data by means of answering question.

In order to maintain consistency between English and Cantonese, the meaning of recorded answers in the two languages must be accordant. For example, if the answer to "What is your favorite color?" in English is "purple", then the meaning of the answer in Cantonese should be related to "紫色". There were 10 such bilingual questions in each enrollment session and 6 in the verification session.

Furthermore, there were 18 tasks designed for the speakers in each enrollment session and 7 different tasks in the verification session. Speakers were required to use their own words to give commands for execution. An example of recording data by means of providing commands for executing tasks is shown in Table 3.2.

Task:	Open the door.	开门.
Short Command:	Open.	开.
Medium Command:	Open the door.	开门.
Long Command:	Please open the door for me.	给我开门.

Table 3.2: Example of recording data of Commands

For each answer or command in the two languages, the speaker is required to provide three versions of different length; "short", "medium", and "long". Examples are shown in Table 3.1 and Table 3.2. The objec-

tive of this procedure is to ensure the data is more robust with regard to text-independency.

Therefore, the composition of the enrollment data and the verification data can be summarized as in Table 3.3.

No. of utterances	Enrollment data	Verification data
Answers	10	6
Commands	18	7
Versions	3	3
Sessions	3	1
Languages	E, C	E, C
In total	504	78

Table 3.3: Composition of enrollment data and verification data for each speaker. (The three versions of data include short, medium and long versions of answers and commands. 'E' refers to English. 'C' refers to Cantonese.)

3.2.2 Data Collection Process for the CUBS Corpus

The speech data of the CUBS corpus was recorded in an office environment. The recording set was derived from a corner in a laboratory. We did not deliberately avoid environmental noise, such as the sounds emanating from the air conditioning system and multiple computer fans. Voices from other speakers could also be heard at various times. A high-quality microphone (SHURE BG1.1) was used to collect the speech data. The waveforms were sampled at 16 kHz.

The recording interface was developed by HTML embedded with Javascripts. Javascripts were used to transfer recording commands to software, which can record speech through a microphone connected to

a computer. An example interface is shown in Figure3.1.

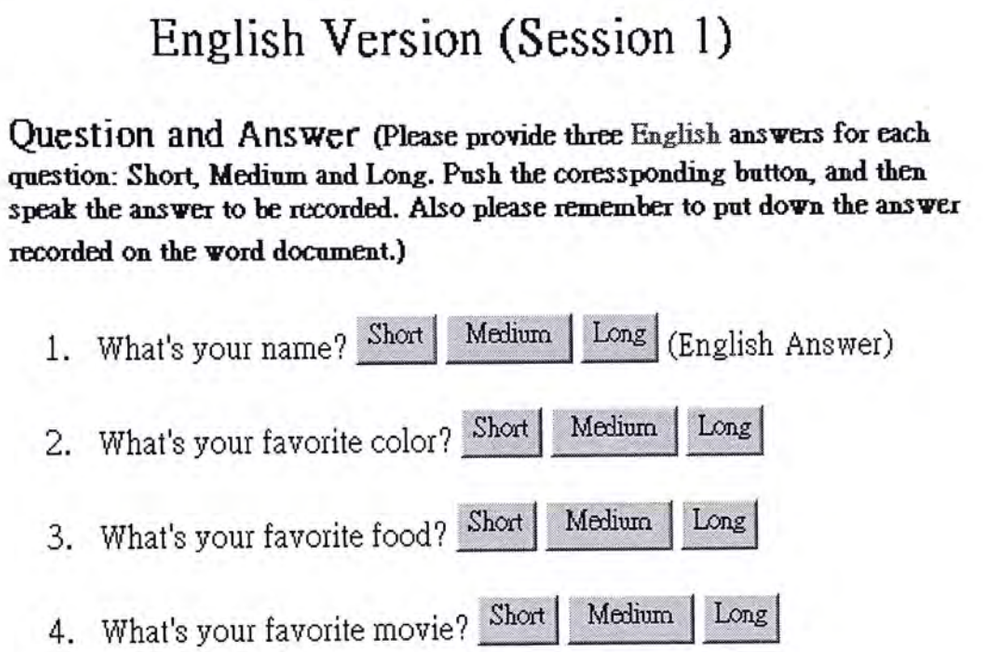

Figure 3.1: An example of data collection interface for the CUBS corpus.

To record answers or commands, speakers pushed the appropriate button on the interface according to the question or task. Recording began automatically. "Short", "Medium"and "Long"refers to the length of the answers and commands. When speakers completed their utterance, a silence detection function built into the recording software automatically stopped the recording.

Since the collection of speech data included three enrollment sessions and one verification session, the whole period of data collection spanned one and a half months. There was a one-week interval between enrollment sessions. The questions and tasks used to prompt responses for recording were the same for all enrollment sessions. Data from the verification session was collected several days after the collection of data from the first enrollment session.

In order to maintain consistency in all sessions, a text record of the speaker's profile was maintained according to the responses collected during the first enrollment session. This recording was then used in the second and third enrollment sessions. This prevented speakers from providing different answers or commands in different sessions, thus ensuring that a user's initial response (for example, purple in response to the question about favorite color) was consistent across all sessions.

3.3 Chapter Summary

In this chapter, we introduced the YOHO corpus and the CUBS corpus. The YOHO corpus is a vocabulary-constraint corpus. It is designed for testing the prototype text-dependent speaker verification system. The CUBS corpus was developed by the author for the purpose of testing a bilingual text-independent speaker verification system. The speaker verification experiments based on these two corpus will be discussed in the following two chapters.

□ **End of chapter.**

Chapter 4

Text-Dependent Speaker Verification

In this chapter, we will explore the foundation of text-dependent speaker verification systems based on the YOHO corpus (Section 3.1). Many different techniques can be involved in the investigation of speaker verification, as outlined in Chapter 2. In order to develop a reliable SV system, we need to choose a logical combination of various techniques. In order to do this, we compare two feature extraction setups based on MFCC and LPCC respectively. The more robust of the two will be used in further experiments. The suitability of two statistical speaker modeling techniques, HMM and GMM, are investigated for speaker modeling in the TDSV system. The cohort normalization technique will also be applied and tested, so as to compensate for the variation in speakers' voice characteristics.

In particular, we describe the front-end processing setup of our experiments in Section 4.1. Based on the introduction to the cohort normalization in Section 2.2.5, the concrete cohort normalization setup

for our SV experiment is described in Section 4.2. Specialized system evaluation methods to evaluate our SV experiments are also briefly described in Section 4.2. Section 4.3 presents SV experiments using HMM modeling techniques. The effect of cohort normalization will also be discussed in Section 4.3. On the basis of an HMM-based SV experiment, GMM-based SV is investigated in Section 4.4. The GMM setup and the comparison of HMM and GMM are also presented in this section. Finally, we compare the performance of our SV system with a number of other systems in order to show that our SV system performs satisfactorily.

4.1 Front-End Processing on the YOHO Corpus

Since the speech data of the YOHO corpus was collected using a high quality telephone handset, the speech utterances from the corpus are first digitized with a bandpass filter ranging from 300Hz to 3400Hz. The digitized data was then pre-emphasized with a first-order difference digital network. Following this, we were able to convert the pre-emphasized data to the feature vector coefficients every 10ms over 25.6ms Hamming windows for each utterance.

Many types of feature representations have been used in speaker verification systems [10] [15] [29] [37] [17]. In our first experiment, we compare two of the most popular - LPCC and MFCC. Results show that the feature parameters of our baseline system are more robust. This superior feature-parameter setup will be used in further experiments. The comparison of the LPCC-based and MFCC-based feature extraction setups is shown as Figure 4.1.

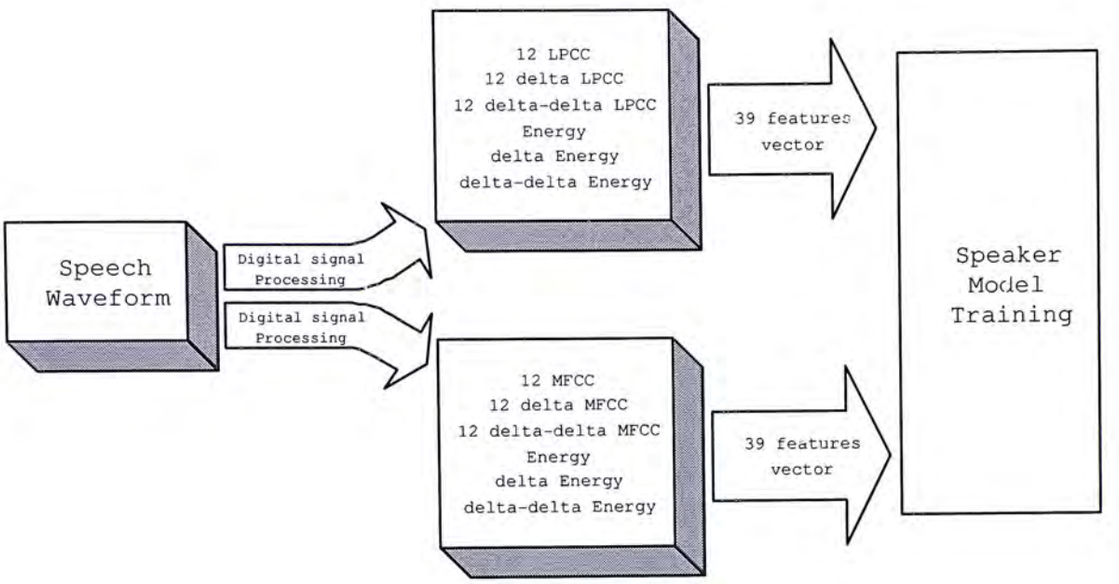

Figure 4.1: Comparison of two kinds of feature representation setups (LPCCs VS MFCCs).

For LPCC feature vectors, the pre-processed speech utterances are first converted into 14th order LPC coefficients every 10ms over 25.6ms Hamming windows. The first 12th order LPC cepstral coefficients are converted from these LPC feature coefficients. Combined with the signal's log-energy, there is a set of 13 acoustic feature coefficients. These 13 feature coefficients will be augmented with their derivative delta and delta-delta parameters. This setup leads to a total of 39 coefficients for each vector.

A feature vector dimension of the same size is used in the MFCC-based experiment. 12 MFCCs are obtained from a bank of 20 filters that are arranged along the Mel-scale over the 300Hz to 3400Hz band. The signal's log-energy is also used. Similarly, the 12 MFCCs and one log-energy coefficient are augmented with their derivative coefficients. Hence, there are 39 coefficients in total for each vector in the MFCC-based feature extraction setup.

4.2 Cohort Normalization Setup

We use the cohort normalization technique in our speaker verification system in order to compensate for variations in speakers' voice characteristics. The cohort normalization approach uses the scores from a group of speakers' utterances to normalize the true scores for speakers' utterances. This group of speakers are determined to be closest to or most "competitive" with the true speaker. In this section, we will begin by presenting our approach to cohort normalization. Based on this setup, we will then describe the techniques for evaluating our TDSV system.

Since there are data for 106 male speakers and only 32 female speakers in the YOHO corpus, the cohort normalization setup for male speakers will be described in detail. First, in order to avoid using a claimed speaker to perform normalization, the 106 male speakers within the YOHO corpus will be separated into two sets. One set consists of test speakers and the other consists of cohort speakers. Among the 106 male speakers, 25 are randomly selected to form the cohort speaker set, while the remaining 81 male speakers will form the test speaker set. The speakers in the cohort speaker set will not be used as impostors or testing the true speaker model. As shown in Figure 4.2, if one speaker is selected to be the true speaker in the test speaker set, all the other 80 speakers will be treated as impostors. Given these parameters, the distance between the true speaker's model and the 25 cohort speakers' models is calculated using Eq.2.28 ($d(\lambda_a, \lambda_b) = \log \frac{p(O_a|\lambda_a)}{p(O_a|\lambda_b)} + \log \frac{p(O_b|\lambda_b)}{p(O_b|\lambda_a)}$). The closest K ($K \leq 25$) cohort speakers will be assigned to the cohort of the true speaker. As we

described in Section 2.2.3, the test utterances from the true speaker are scored against his own models and the associated cohort models. The final likelihood ratio scores for each utterance are calculated using Eq.2.27 ($\Lambda(O) = \log p(O|\lambda_C) - \frac{1}{K}\sum_{k=1}^{K} \log p(X|\lambda_k)$). Thus, the FRR curve can be plotted according to those scores. Similarly, the test utterances from the other 80 speakers are also scored against the true speaker's models and his cohort speakers' models, so as to plot the FAR curve. Finally, the EER for this speaker can be found at the point where the FRR curve and the FAR curve intersect.

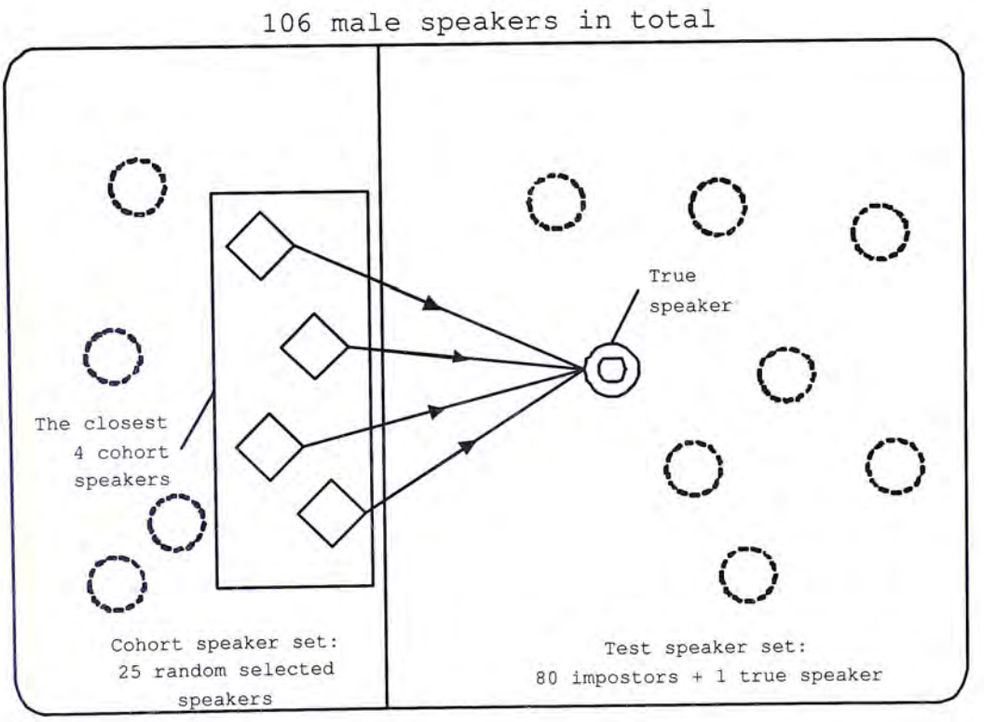

Figure 4.2: Setup of cohort normalization.

The true speaker in our experiment is selected from within the test speaker set in rotation. Once a true speaker is selected, all the other 80 speakers will be treated as impostors. The K closest cohort speakers will be assigned to the true speaker from the cohort speaker set. This

procedure will be repeated N_{spk} times, where N_{spk} is the number of speakers within the test speaker set. For example, N_{spk} equals to 81 for male speakers. Therefore, it is possible to obtain N_{spk} EERs, where the mean EER (Eq.4.1) will be taken as the evaluation value of the verification system. This approach is the same as described by Yoik [9].

$$Mean\ EER = \frac{1}{N_{spk}} \sum_{i=1}^{N_{spk}} EER_i \tag{4.1}$$

Since we use a randomization process to select the speakers who form the cohort set, the quality of the 25 speakers in the cohort set may bias speaker verification performance. Therefore, we repeat the whole procedure 10 times, from the cohort set random selection to the *mean EER* calculation. Then we take the average (Eq.4.2) of the ten *mean EER*s as the final result to evaluate our system, so as to compensate for the effects of a randomly selected cohort speaker set.

$$\overline{EER} = \frac{1}{10} \sum_{j=1}^{10} (Mean\ EER)_j \tag{4.2}$$

Thus, the final experimental result we use to evaluate system performance is obtained through two averaging procedures. Using the $\overline{EER}$ to evaluate our SV system leads to more accurate experimental results, because the bias inherent in randomly selecting speakers from the cohort set (Figure 4.2) can be effectively reduced.

A similar setup of cohort normalization is applied to female speakers. Because there are only 32 female speakers in the YOHO corpus, 15 are selected randomly to constitute the cohort set, and the remaining 16 constitute the test speaker set.

4.3 HMM-based Speaker Verification Experiments

This section describes the experimental setup of an HMM-based TDSV system. Our results and analysis are based on two aspects: a comparison of feature parameters, and the effect of cohort normalization.

4.3.1 Subword HMM Models

In our HMM-based TDSV system, speakers are represented with a set of left-to-right HMMs with continuous density Gaussian mixtures. These HMMs represent the spoken digits and non-speech segments from the YOHO corpus speech data by means of modeling different types of acoustic units (Section 2.2.2.1). Acoustic units here refer to phone-like units, "phonemes" and acoustic segment units, "subwords", and so on. We construct a set of subword HMMs for each of the "digits" and "decades" in the YOHO corpus. The HMMs are summarized in Table 4.1. Each subword HMM contains eight Markov states, with each state containing four Gaussian mixture components.

The silence model "sil" is used to model non-speech utterances. We use a three-state left-to-right HMM with four mixture components per state to estimate "sil" (Figure 4.3). An extra transition from state 2 to state 4 is added in the silence model. The reason for doing so is to make the model more robust by allowing individual states to absorb impulsive noise into the training data. A backward skip allows for this without committing the model to transit to the following word.

Since there may be short pauses between utterances consisting of two doublets, a short pause model "sp" is employed to model this pause. A one-state "tee-model" [50], which includes a direct transition

Digit	Model	Decade	Model
1	one	20	twenty
2	two	30	thirty
3	three	40	forty
4	four	50	fifty
5	five	60	sixty
6	six	70	seventy
7	seven	80	eighty
9	nine	90	ninety
short pause	sp	silence	sil

Table 4.1: List of subword HMM models in the YOHO database.

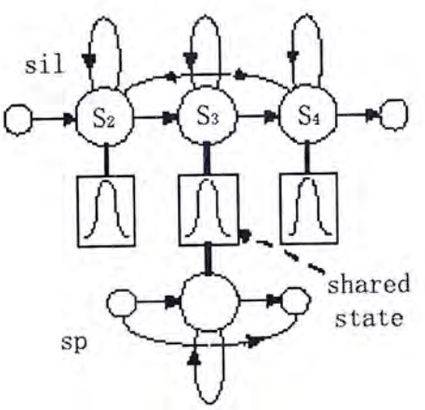

Figure 4.3: Silence model and short pause model.

from the entry to the exit node, will be treated as "sp". The emitting state of the "sp" model is tied to the center state of the silence model. The topology of both HMMs, which are used to model the non-speech segment of utterances, is shown in Figure 4.3.

Each speaker is represented by a set of speaker-dependent HMMs as shown in Table 4.1. The HMMs are estimated by the training data from all of the four enrollment sessions. Each model is initialized with one Gaussian mixture per state. The number of mixture components is increased after the HMMs run on several iterations of Baum-Welch reestimation [50], and reestimation is saturated. The mixture increment is performed by a segmental K-means algorithm. Hence, the increment process is one to two, and then two to four. This means that the increment of mixture components is faster and more reliable. Finally, we obtain a set of well-estimated speaker-dependent HMMs for each speaker.

4.3.2 Experimental Results

All of the speakers (106 males and 32 females) within the YOHO corpus are involved in our experiments. Experiments do not contain inter-gender tests, which means that male speakers are only tested by male speakers and female speakers are only tested by female speakers. This approach is suggested by [5].

4.3.2.1 Comparison of Feature Representations

First, we compare the feature representations of MFCC and LPCC. The comparison of the experimental results is shown in Table 4.2 and Figure 4.4

EER(%)	Male speakers	Female speakers
LPCCs	1.21	2.82
MFCCs	2.23	3.11

Table 4.2: Feature parameters comparison of speaker verification on male and female speakers for the YOHO corpus.

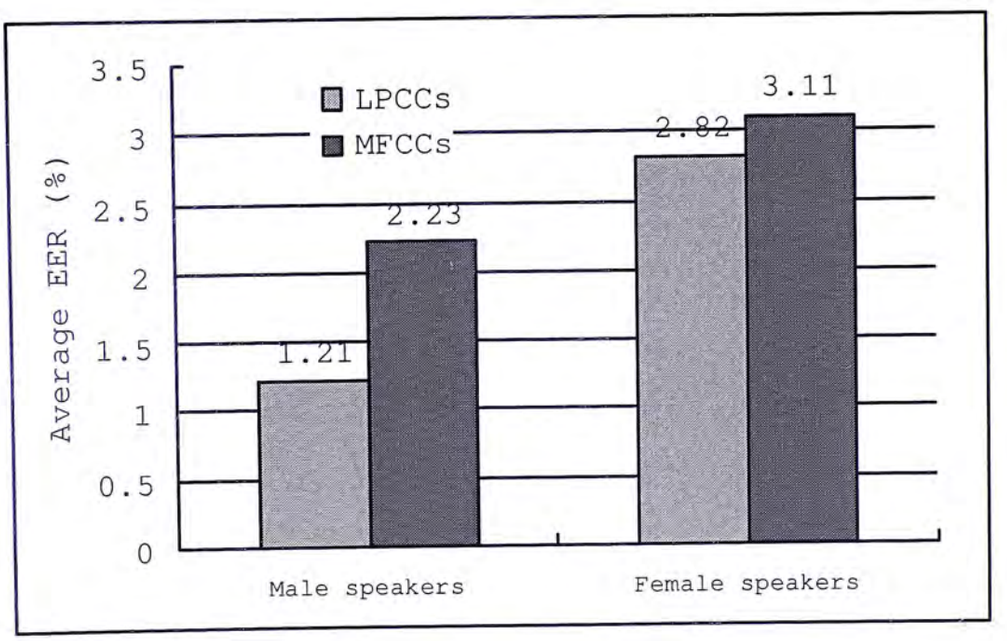

Figure 4.4: Feature parameters comparison of speaker verification on male and female speakers for the YOHO corpus.

The results show that LPCCs outperform MFCCs when they are used as the main feature parameters, for both male and female speakers. For male speakers, LPCCs (EER = 1.21%) outperform MFCCs (EER = 2.23%) 46%. Similar results are obtained in experiments based on female speakers; LPCC (EER = 2.82%) outperforms MFCC (EER = 3.11%) by around 9%.

Previous work [3] [22] has also shown that LPCCs outperform MFCCs in certain circumstances, and especially for the YOHO corpus. LPCCs outperform MFCCs in our experiments because they provide better performance at lower vector dimensions (first 12th order). This is because speaker-related information is more concentrated at the lower order LPCC parameters [22]. Moreover, the non-linear triangular filter bank used in MFCC generation may cause speaker's information to be lost in the higher frequency band, while LPCC's generation treats all frequency bands as the same. When the speech signal quality is sufficient, the prediction error of LPC is small. Therefore, certain LPC-based combinations have the potential to be more effective than FFT-based methods (using MFCC parameters) in improving recognition performance. Based on this result, we adopt LPCC as the major feature representations for feature extraction in the experiments that follow.

It should be noted here that although optimal feature extraction is a very large topic in speaker recognition research, a detailed investigation oversteps the scope of this thesis. A deeper investigation of this topic will be the subject of future work.

4.3.2.2 Effect of Cohort Normalization

We apply cohort normalization to our speaker verification experiments. Different numbers of cohort speakers are tested to ensure normalization. As per the normalization setup described in Section 4.2, the K closest cohort speakers are selected for normalization. For male speakers, $K = 0, 5, 10, 15, 20, 25$. For female speakers, $K = 0, 5, 10, 15$. Here, $K = 0$ means that we have not applied cohort normalization. When no cohort normalization is applied, speaker verification experiments will utilize 81 male speakers or 17 female speakers from the test speaker set (Section 4.2).

Since it has been determined that feature parameters utilizing LPCCs in the main outperform MFCCs (Section 4.3.2.1), we use LPCCs as the major feature in the following experiments. The results of speaker verification using cohort normalization are shown in Table 4.3 and Figure 4.5.

EER(%)	$K = 0$	$K = 5$	$K = 10$	$K = 15$	$K = 20$	$K = 25$
Male	1.21	0.62	0.46	0.52	0.60	0.71
Female	2.82	1.15	0.84	0.91	-	-

Table 4.3: Results of applying cohort normalization for speaker verification with the YOHO corpus.

It is obvious that cohort normalization is very useful for improving verification performance. In the case of male speakers, when $K = 10$, the $\overline{EER}$ is 0.46%, compared with 1.21% when no cohort normalization is applied. The verification performance improves by 2.6 times. Similarly, for female speakers, the $\overline{EER}$ reduces from 2.82% to 0.84%

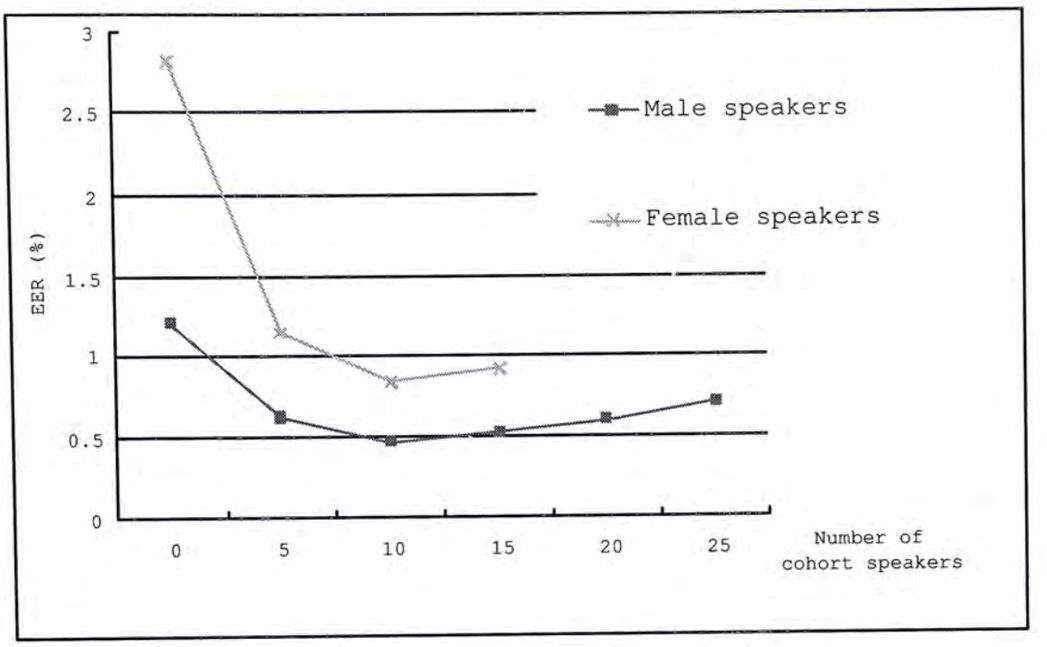

Figure 4.5: Effect of cohort normalization for speaker verification with the YOHO corpus.

when $K = 10$. The verification performance improves 3.4 times.

These results prove that cohort normalization can effectively improve the performance of a speaker verification system. The improvement can be explained by calculating the likelihood ratio. The likelihood ratio is based on the likelihood scores of the true speakers and the cohort speakers. As mentioned in Section 2.2.5, the average likelihood scores of K cohort speakers (Eq.2.26: $\frac{1}{K}\sum_{k=1}^{K}\log p(X|\lambda_k)$) is used to represent the likelihood score of an "ideal" impostor. When a test utterance emanates from a true speaker, the normalized likelihood ratio (Eq.2.27: $\Lambda(O) = \log p(O|\lambda_C) - \frac{1}{K}\sum_{k=1}^{K}\log p(X|\lambda_k)$) will rise. When the utterance emanates from an impostor, the normalized likelihood ratio will decrease. Thus, cohort normalization allows the speaker verification system to distinguish true speakers from impostors

more accurately via the likelihood ratio, which can compensate for the degradation of speaker verification performance caused by variations in a speaker's speech characteristics.

Our experimental results show that there is an optimal K for both male and female speakers. Verification performance is best when $K = 10$. If K is less or larger than 10, the EERs will increase. This phenomena is shown in Figure 4.5. The reason for this is as follows. From the point of view of normalization, the best verification performance can be obtained when the normalization term (Eq.2.26) represents the most aggressive impostor. The most aggressive impostors are those who exhibit speech traits similar to a true speaker. In our experiment, the cohort set is randomly selected from the whole corpus (25 of 106 male speakers and 15 of 32 female speakers). In the cohort set, there must be some speakers similar to the true speaker and others who are dissimilar. K most similar speakers of the true speaker in this set are selected to form the normalization speaker set. Speakers' scores in the normalization set are used to calculate the normalization term in Eq.2.26. Hence, when K is selected optimally, the effect of normalization is greatest. Otherwise, when K is smaller than the optimal number, the normalization set is not able to represent information from cohort speaker sufficiently. If K is larger than the optimal number, dissimilar speakers will constitute the normalization set. Under these conditions the normalization performance degrades. Hence, there is an optimal K for the cohort normalization in our experiments.

4.4 Experiments on GMM-based Speaker Verification

Experiments on GMM-based speaker verification systems will be described in this section. We begin by introducing our experimental design. Following this, we investigate the effect of the number of Gaussian mixture components in a GMM according to the experimental results. Next, we briefly discuss the performance of cohort normalization on GMM-based SV. Finally, we compare HMM-based and GMM-based SV systems.

4.4.1 Experimental Setup

Since the GMM-based SV system has been upgraded from the HMM-based TDSV system (as explained in Section 4.3), we have inherited a number of components from the HMM-based TDSV system. They include speech data processing, feature extraction, scoring and cohort normalization setup, and so on. Because we proved LPCCs outperform MFCCs, the LPCC-based feature extraction setup is used in this experiment. $K = 10$ has proven to be the optimal number of cohort speakers, and is thus used as a standard setup in GMM-based SV experiments.

Generally, the GMM speaker modeling technique is used for TISV. GMM-based speaker verification on the YOHO corpus can be viewed as a vocabulary-constrained speaker verification task. This means that a speaker's GMM only need model a constrained acoustic space. Thus, it allows an inherently text-independent model to be used in a text-

dependent task [34]. As we described in Section 2.2.2.2, each enrolled speaker in our GMM-based speaker verification system will be represented by only one GMM model. The underlying acoustical speech classifications within a single speaker model will be characterized in this model. In our experiments, we use "*speech*" to denote this. The likelihood scores for test utterances are calculated from this single speaker-dependent GMM.

4.4.2 The number of Gaussian Mixture Components

Determining the number of Gaussian mixture components M required to model a speaker adequately is an important but difficult problem. On one hand, choosing too few mixture components can produce a speaker model that does not accurately model the distinguishing characteristics of a speaker's distribution. On the other hand, choosing too many components may reduce performance when the available training data is inadequate to train a large number of model parameters. Moreover, doing so will lead to excessive computational complexity in both training and testing. However, there is no theoretical method to guide us in finding an "optimal" solution. Therefore, our aim is to find the "optimal" number of Gaussian mixture components M empirically.

All of the training data (96 utterances over 4 sessions) for each speaker in the YOHO corpus are used to train a speaker's GMM model. First, a one-state GMM with one mixture component is initialized. The number of mixture components M will be increased after several Maximum Likelihood (ML) reestimations (5 times in our experiment). In order to find the optimal M, M is increased geminately from 1 to 2, 2 to 4, ..., 64 to 128, until the verification performance begins to

drop. In this way, the optimal number of Gaussian mixture components can be obtained. Previous work [34] has shown that the YOHO corpus contains enough data to train 64 Gaussian mixture components. Hence, in our work we compare the performance of GMMs with 32, 64 and 128 mixture components for speaker verification. The experimental results are shown in Table 4.4 and Figure 4.6.

EER (%)	32 components	64 components	128 components
Male speaker	1.59	0.89	4.22
Female speaker	3.84	2.03	5.66

Table 4.4: The effect of different numbers of Gaussian mixture components on TDSV.

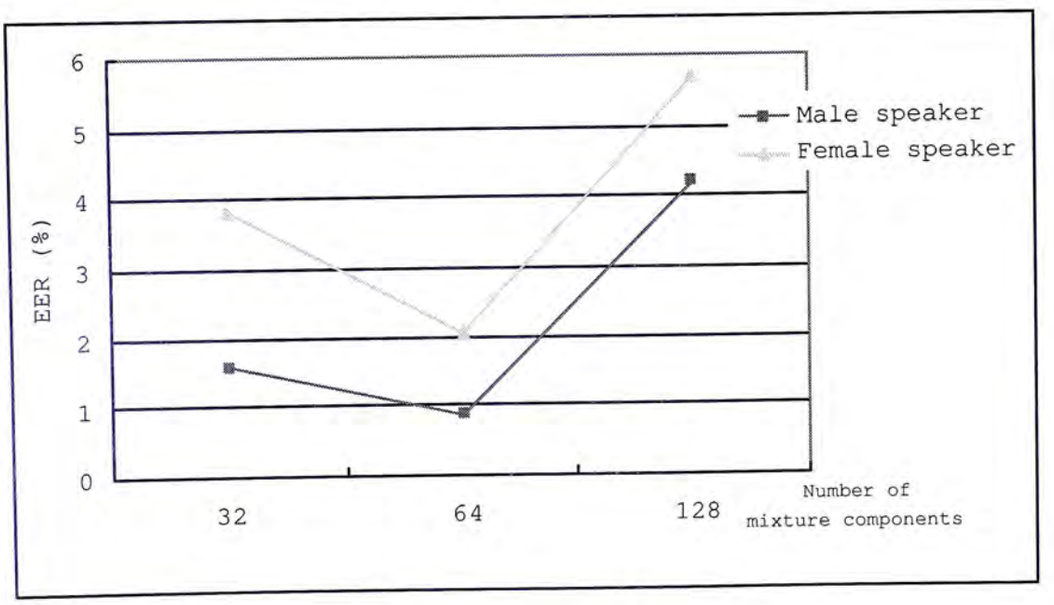

Figure 4.6: the effect of different numbers of Gaussian mixture components on TDSV.

The best performances regarding male and female GMM-based speaker verification are 0.89% and 2.03% respectively, and when the GMM has 64 mixture components. The experimental results show that by us-

ing the available amount of training data in the YOHO corpus we can train a better speaker-dependent GMM with 64 mixture components. When we increase M from 64 to 128, the EER increases from 0.89% to 4.22%. The verification performance dropped greatly because the data are inadequate to train a larger number of mixture components satisfactorily in a GMM.

These results prove that determining the relationship between the number of mixture components M and the amount of training data is important for improving speaker verification performance. Therefore, it is of considerable importance to select an appropriate M according to the amount of training data. Too much training data in combination with a small number of mixture components will lead to overtraining. The main problems associated with overtraining are that, on the one hand, a number of distinctive characteristics cannot be described and are thus confused with other characteristics. On the other hand, if M is too large in relation to the amount of training data, each mixture component is not able to adequately represent the distribution of data. In both cases, the GMM cannot model training data characteristics. This failure results in a reduced ability to distinguish between speakers, and hence speaker verification performance is degraded.

4.4.3 The Effect of Cohort Normalization

Having applied cohort normalization to our GMM-based SV experiment, we are able to determine the level of improvement obtained by using this technique. Table 4.5 records the effect of cohort normalization when M is 64 in GMM.

As the table shows, we can determine that cohort greatly improves

EER(%)	K=0	K=5	K=10	K=15	K=20	K=25
Male speakers	5.39	1.11	0.89	0.91	0.99	1.58
Female speakers	7.87	2.55	2.03	2.08	-	-

Table 4.5: The Effect of Cohort Normalization on GMM-based TDSV (M = 64).

the performance of a GMM-based speaker verification system. Our results show that when K is 10, the EER decreases from 5.39% to 0.89% (about 6 times) for male speakers. For female speakers, the EER decreases from 7.87% to 2.03% (about 4 times). The results show that cohort normalization can also effectively compensate for variations in GMM-based speaker verification; therefore, applying this technique improves verification performance.

4.4.4 Comparison of HMM and GMM

In GMM-based TDSV experiments, we have used a single GMM to substitute for subword HMMs in an HMM-based TDSV. This section compares HMM- and GMM-based speaker verification. Since we know that $M = 64$ is optimal number of Gaussian mixture components, we use a single-state GMM with 64 mixture components for speaker modeling in this GMM-based SV experiment. Doing this enables us to compare the performance of HMM-based TDSVs and GMM-based TDSVs. The comparative results of our experiments are displayed in Table 4.6 and Figure 4.7.

Experimental results show that an HMM-based TDSV outperforms a GMM-based TDSV by 48% and 59% for male and female speakers respectively. This result can be explained with reference to text-

EER(%)	Male speaker	Female speaker
HMM-based SV	0.46	0.84
GMM-based SV	0.89	2.03

Table 4.6: Performance Comparisons of HMM- and GMM-based speaker verification.

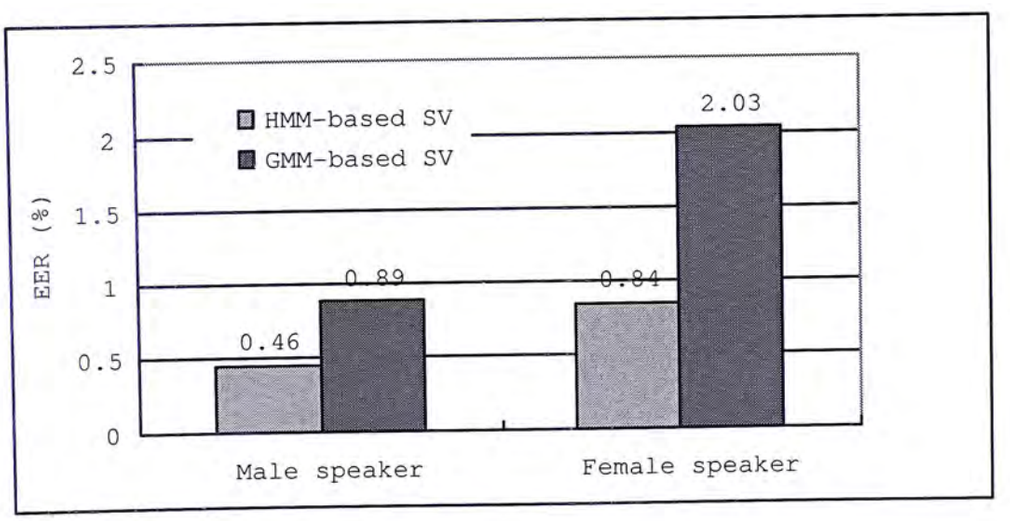

Figure 4.7: Performance Comparisons of HMM- and GMM-based speaker verification.

dependency. In the HMM-based TDSV system, subwords are the basic unit of the acoustic model, Therefore, speaker characteristics are described in a more detailed acoustic space than is available in the GMM-based TDSV system. If the amount of training data is sufficient, detailed acoustic models trained sufficiently well have the capacity to describe speaker-specific characteristics in a higher resolution than is possible in a GMM. This is because GMMs combine all of a speaker's information into a single model without classification. Phonetic information from the training and testing data determines the classification of the acoustical model units in the HMM-based TDSV. However, GMMs discard phonetic information during training and testing. Because of this, GMMs may lose speaker characteristics based on detailed acoustic-units. Thus, HMM-based SV systems can outperform GMM-based SV system in similar setups.

4.5 Comparison with Previous Systems

There are many reports on the performance of speaker verification systems using the YOHO corpus. A comparison of our system's performance and the performance of other systems is shown in Table 4.7 and Figure 4.8. Che's [7], AT&T's [43] and Cheng's [9] SV systems are HMM-based. MIT's [34] SV system is GMM-based. MIT's GMM-based SV system used 64 Gaussian mixture components. All these SV systems used enrollment data from all four sessions in the YOHO corpus to train speaker-specific models. The differences between these systems are mainly due to two reasons. The first one is that different kinds of modeling techniques and HMM units are used to model

speech segments. The other important reason is that different training or testing paradigms and cohort speaker sets are used in SV systems.

EER(%)	Male speaker	Female speaker
Che (HMM)	0.09	0
AT&T (HMM)	0.47	N/A
Cheng (HMM)	0.48	N/A
Mine (HMM)	0.46	1.15
MIT (GMM)	0.20	1.88
Mine (GMM)	0.89	2.03

Table 4.7: Comparison of our speaker verification performance with other systems.

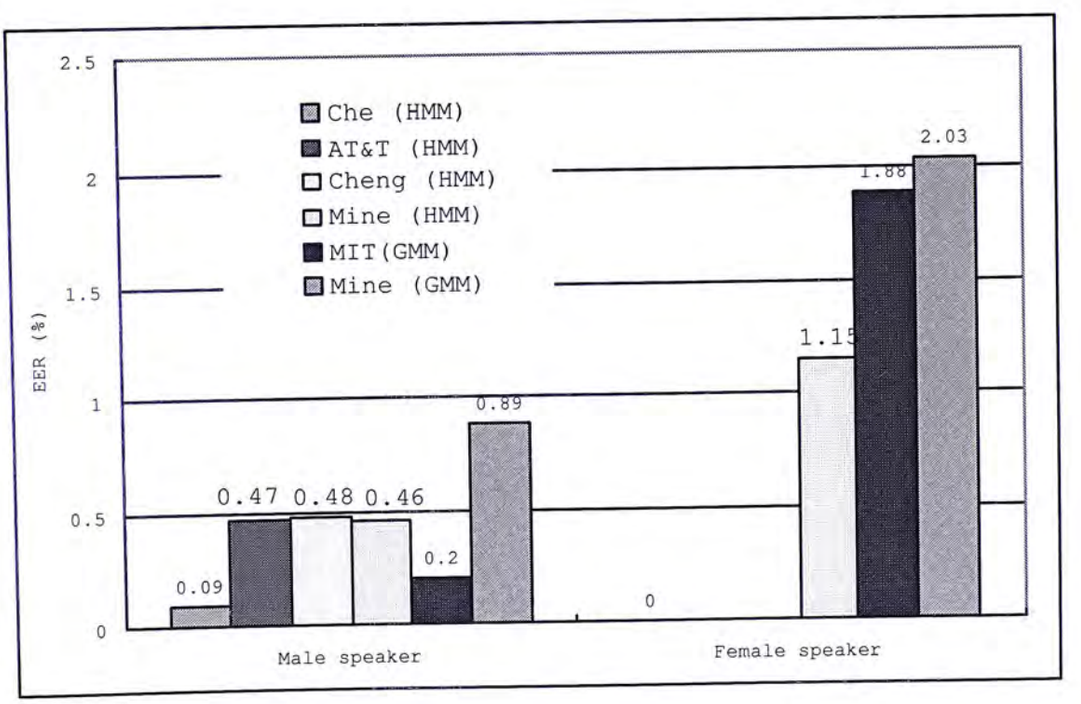

Figure 4.8: Comparison of our speaker verification performance with other systems.

For example, Che's HMM-based SV system uses phonemes as the

basic HMM unit. Che's SV system outperforms all others because it uses additional training data. All phoneme-based HMM units in Che's system are bootstrapped using parameters from the Resource Management and TIMIT databases. The additional training data is used to train the parameters of the speaker-independent phoneme HMM seeds. Subsequently, the seeds are trained by speaker specific data in the YOHO corpus. These bootstrapped phoneme-based HMMs then develop into speaker-dependent models. The advantage of using additional training data is that the TDSV system can reduce errors caused by the recognition of speech segments.

Che's HMM-based and MIT's GMM-based SV systems outperform all other systems. To a certain extent, this is due to their specific testing paradigms. The testing trials in both Che's and MIT's SV systems consists of all four utterances in any given speaker's verification session. Each speaker performs 10 verification tests against himself/herself. Including a greater number of utterances in a trial will reduce false probability. For example, there are 40 utterances over 10 verification sessions for each speaker. To calculate FRR using one utterance per trial, when the likelihood score of one true speaker's utterance is lower than the threshold, the FRR registers 2.5%. However, although calculating FRR using four utterances per trial reduces the likelihood that one true speaker's utterances are lower than the threshold, it improves the likelihood scores of the other three utterances' in that trial. Hence, the FRR may be still 0%. This explains why Che's HMM-based and MIT's GMM-based SV systems outperform others.

There may also be some other unknown techniques used in these SV systems not mentioned in the reports. These unknown factors may

also account for differences in performance. Hence, these results can only be loosely compared.

With a similar experimental approach, we can observe similar performance between our system, AT&T's, and Cheng's HMM-based SV system. For experiments with male speakers, our SV systems' EERs are 0.46%, 0.47% and 0.48%. These results prove that our SV baseline is reliable.

4.6 Chapter Summary

In this chapter, we described the foundation of our HMM-based and GMM-based speaker verification systems. We compared two kinds of feature representations, LPCCs and MFCCs. We explained why LPCC is a better approach for feature extraction, and thus chose it for further experiments. Our experimental results proved the cohort normalization effect. More than 60% improvement can be obtained when this technique is implemented. The comparison of HMM-based and GMM-based TDSV is presented. In text-dependent scenarios, HMM-based SV system outperforms GMM-based SV systems. In comparison with other previous YOHO-based SV systems, we proved that our speaker verification baseline is reliable. It provides a solid foundation for further investigation on language-independent and text-independent speaker verification.

□ **End of chapter.**

Chapter 5

Language- and Text-Independent Speaker Verification

To verify a person's identity without language restrictions is a great advantage for speaker verification systems. On the other hand, a TISV system can be integrated into a dialog system. Although its performance may be somewhat lower than TDSV, a speaker's private information can be used to enhance its robustness within a dialog system [20] [23] [19]. It is obvious that language-independence can provide much flexibility to the verification process.

However, there is limited work focusing on the effect of language-dependency in speaker verification. Previous work [2] has examined the dependency of specific language models in the TISV system. The results showed that TISV performance is affected by specific languages, namely Arabic, American English, Farsi, French, German, Hindi, Japan-

ese, Korean, Mandarin, Spanish, Tamil and Vietnamese. Experimental results have demonstrated that Vietnamese and Mandarin behave differently from English. Another empirical study on the use of GMMs for multilingual (English, Mandarin and Cantonese) speaker verification can be found in [31]. Qing and Chen proved that it is possible to use GMMs as a model for multilingual speaker verification by way of a number of simple experiments.

Therefore, based on our previous investigations of HMM- and GMM-based speaker verification, this chapter will systematically discuss the language-dependency effect on TISV. Our research focuses on the language-dependency of English and Cantonese in particular. From this, a bilingual TISV system is developed. The work is evaluated by the CUBS corpus, which has been developed specially for this purpose.

In Section 5.1, a brief introduction to the front-end processing of the CUBS corpus is compared with the YOHO corpus. The reason for using GMM to investigate the bilingualism of a TISV system is presented in Section 5.2. Cohort normalization is still an essential component of our speaker verification system, and its setup is described in Section 5.3. Finally, the reports on the experimental results, including the selection of the number of Gaussian mixture components, the language-dependency investigation, and the language-independent speaker verification development are presented and analyzed.

5.1 Front-End Processing of the CUBS

In order to investigate the language-independent TISV, we replace the YOHO corpus with the CUBS corpus, which was developed with bilin-

gual text-independent speech. Since the data from the YOHO corpus is simulated telephone quality speech, it is sampled at 8 kHz (Section 3.1). The data from the CUBS corpus is recorded by desktop microphone in a real office environment. It is microphone quality speech data and is sampled at 16 kHz. For this reason, we have had to change the sampling rate from 8 kHz to 16 kHz in front-end processing. Other than the change in sampling rate, we have used the feature extraction setup of our previous system so as to ensure correctness. Therefore, each feature vector includes the first 12th order LPCCs that are derived from the 14th LPC coefficients and the log-energy of the speech signal. In addition, the vector also includes the derivative delta and delta-delta coefficients of the LPCCs and energy coefficients. There are 39 coefficients in total for each feature vector.

5.2 Language- and Text-Independent Speaker Modeling

We have two choices when using statistical methods to solve speaker modeling problems in a language-independent TISV system. One is HMM-based, the other is GMM-based. The HMM-based method requires we find all possible speaker-dependent HMMs with respect to the acoustic units in different languages. That is, if the amount of training data is adequate, we need to train sufficient left-to-right HMMs to model the entire acoustic space. If this is the case, then we can find a combination of the well-trained HMMs to model any utterance made by a speaker. However, it is impractical to collect a large enough amount of training data that would enable us to train all of the speaker-

dependent HMMs. Hence, we propose to test this method when it is possible for us to collect enough speaker-dependent speech data in the future.

Our work has also proved that the GMM is very effective for modeling speaker-specific characteristics in the absence of phonetic information. For example, in the constrained vocabulary speaker verification scenario, GMMs perform very well at modeling a speaker. Although discarding phonetic information in a speaker model will degrade the speaker verification performance (Section 4.4.4), it bring about great flexibility in solving the text-independent and language-independent SV problems. From this point of view, we adopt GMM modeling techniques to build language-independent and text-independent speaker verification system in this investigation.

5.3 Cohort Normalization

We have proved that cohort normalization can greatly improve the performance of speaker verification by compensating for the variations in voice characteristics with regard to the TDSV task (Section 4.3.2.2). Therefore, this technique is still necessary for language-independent TISV. We have made a number of changes to the cohort normalization setup according to the differences between the YOHO corpus and the CUBS corpus. For example, the CUBS corpus has considerably fewer speakers than the YOHO corpus. The process of selecting cohort speakers for each test speaker is shown in Figure 5.1.

First, we chose a single speaker **A** from the entire speaker set as the true speaker. Eight speakers were randomly selected to form the cohort

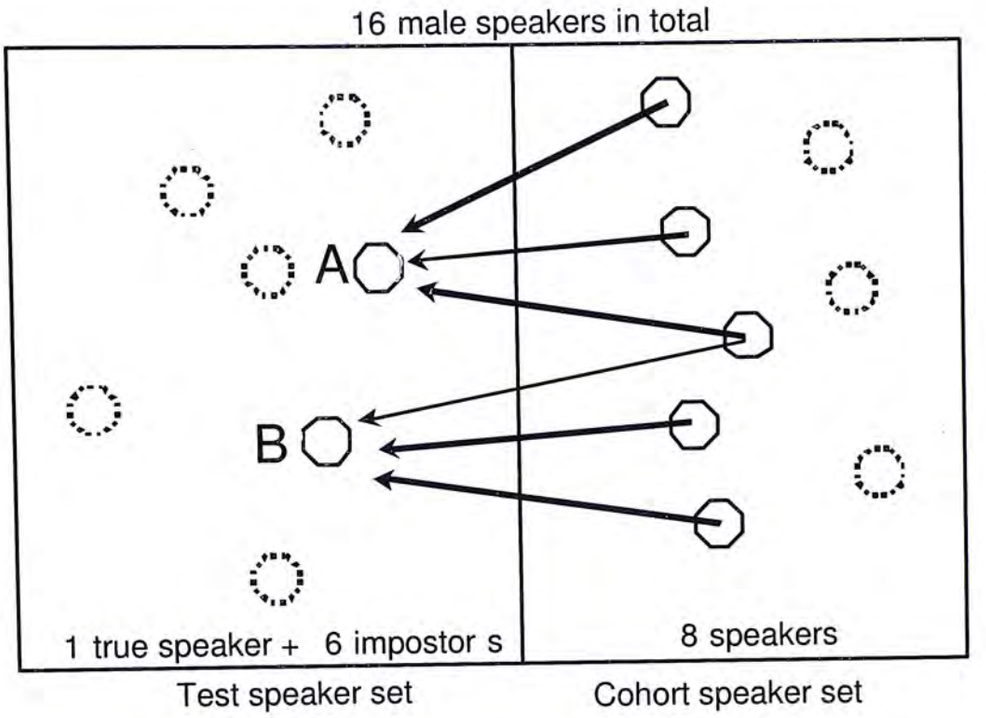

Figure 5.1: Cohort speakers selection.

speaker set for **A**, from which the remaining 6 speakers were regarded as impostors. By doing so, we were able to find the closest K speakers to **A** from among the cohort speaker set using the similarity measure Eq.2.28. K is the number of cohort speakers assigned to speaker **A** (K = 1, 2, 3, 4). Finally, we repeated the process for each speaker and thus obtained the EER for each of them. The mean EER of the 16 speakers was treated as the evaluation criterion for our system.

5.4 Experimental Results and Analysis

Our experimental results and analysis focus on four major points. First, we determine the optimal number of Gaussian mixture components (M) in the speaker-dependent GMM trained with the CUBS corpus. Second, we discuss the effect of cohort normalization on CUBS-based

TISV. Third, we investigate the language-dependency of speaker verification through experiments on English and Cantonese data from the CUBS corpus. Fourth, the bilingualism of a speaker verification system is described and analyzed.

In the first experiment, we used data from all three training and enrollment sessions from the CUBS corpus to train the language-dependent speaker-specific GMM model. In other words, English training data is used to train the English speaker model, and Cantonese training data is used to train the Cantonese model. All testing data from the verification data set was used to test speaker modeling. The testing process is language-dependent. Only English testing data are used to test the English model, and Cantonese testing data to test the Cantonese model. Each trial consists of utterances in the verification data set. Table 5.1 and Figure 5.2 show experimental results using English data from the CUBS corpus. Table 5.2 and Figure 5.3 show the experimental results using Cantonese data from the CUBS corpus. These tables and figures report the optimal number of Gaussian mixture components and the cohort normalization effect.

No. of mixture components	EER of different K cohort speakers (%)				
	No cohort speakers	$K = 1$	$K = 2$	$K = 3$	$K = 4$
64	27.71	7.86	5.41	5.36	5.13
128	24.90	7.88	5.77	4.88	4.08
256	21.10	4.91	4.89	4.11	3.98
512	21.78	5.49	4.69	4.33	4.10

Table 5.1: The effects of mixture component numbers and cohort size in English.

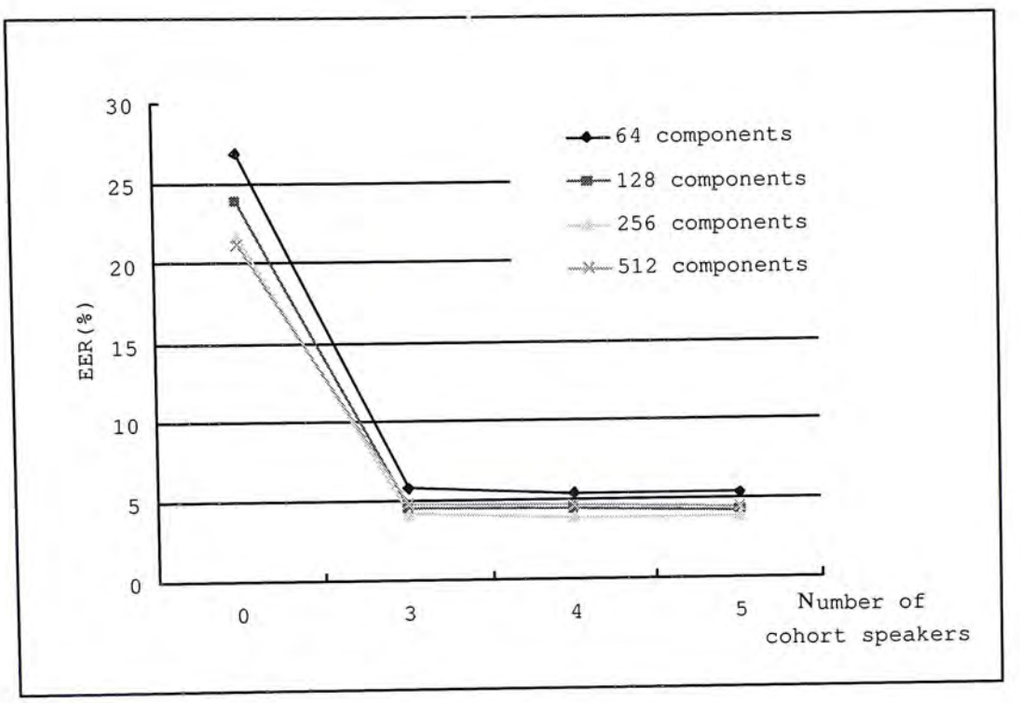

Figure 5.2: The effects of mixture component numbers and cohort size in English.

No. of mixture components	EER of different K cohort speakers (%)				
	No cohort speakers	$K = 1$	$K = 2$	$K = 3$	$K = 4$
64	27.37	7.80	5.68	5.06	5.24
128	24.28	6.18	5.15	4.92	4.37
256	21.01	6.27	4.44	4.20	4.28
512	22.02	5.72	4.95	4.43	4.10

Table 5.2: The effect of mixture component numbers and cohort size in Cantonese.

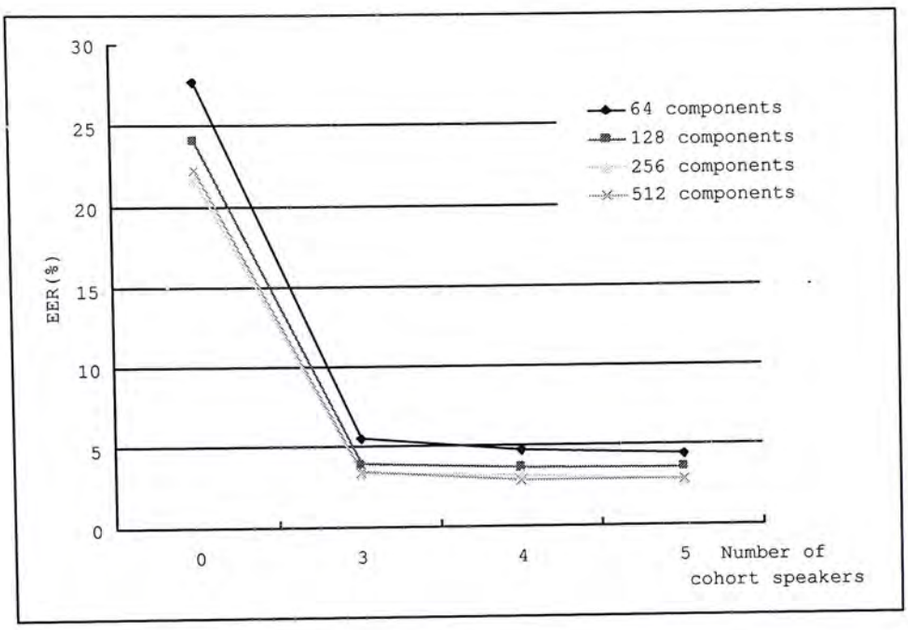

Figure 5.3: The effect of mixture component numbers and cohort size in Cantonese.

5.4.1 Number of Gaussian Mixture Components

As we have found in the GMM-based SV experiments on the YOHO corpus, if the amount of training data is sufficient, a greater number of mixture components within a GMM can lead to better performance in describing a speaker's characteristics in detail. Therefore, we expect to use a larger number of mixtures in the GMM. However, the number of mixture components is constrained by the amount of training data. For training data to be sufficient to train the mixture components well, it is essential that more mixture components exist. If so, a better speaker model can be described.

In these experiments, the K-means algorithm is used to perform mixture splitting. Hence, the number of mixture components M is increases by doubling (e.g., from 1 to 2 and from 2 to 4). We test

the GMM with 64 mixture components first, and increase M step by step. $M = \{64, 128, 256, 512\}$ are tested. The increment stops when the process cannot further improve the system's performance. This occurs when the amount of training data is insufficient to effectively train an increased number of mixture components.

Results show that the parameter setting $M = 512$ cannot outperform that of $M = 256$ in either English and Cantonese experiments. When M is increased from 256 to 512, the EER increases from 21.10% to 21.17% (Table 5.1) in the experiments based on English data. When the number of mixture components increases from 256 to 512, the EER increases from 22.01% to 22.02% (Table 5.2) in the experiments based on Cantonese data.

The results show that the training data in each language is only adequate to train 256 Gaussian mixture components in a language-dependent GMM. If M is increased to 512, the Gaussian mixture component cannot represent the data distribution well and the system performance will degrade. Hence, we use a GMM with 256 mixture components to model speakers in further experiments.

5.4.2 The Cohort Normalization Effect

Another point that can be observed from the results in Tables 5.1 and 5.2 as well as Figures 5.1 and 5.2 is the effect of cohort normalization. When 4 cohort speakers are used in the cohort set ($K = 4$), the highest verification performance is obtained. When 4 cohort speakers are used in experiments using English data, the EER is reduced from 22.10% to 3.98%. When 4 cohort speakers are used in the experiments using Cantonese data, the EER is reduced from 22.01% to 4.28%.

The results show that cohort normalization has a great impact on compensating for the variations of a speaker's voice characteristics. The verification performance improved by 5.6 and 5.1 times respectively when cohort normalization was applied. This is sufficient when we compare the effect of cohort normalization on GMM-based SV experiments with the YOHO corpus.

5.4.3 Language Dependency

Based on our investigations into the number of Gaussian mixture components and cohort normalization for GMM-based text-independent speaker verification, we turn in this section to exploring the language dependency of speaker verification systems with the CUBS corpus. The speaker-dependent GMM is trained with 256 Gaussian mixture components using the data from all three sessions where speakers provided language-dependent data. Cohort normalization is also applied. $K = 4$ is used in all experiments. All testing data in the verification set from the CUBS corpus are used to test speaker modeling.

In order to test language-dependency for English and Cantonese based for the CUBS corpus, we set up an experiments as follows. First, English and Cantonese data were used to train the speaker-dependent English model and Cantonese model respectively. The English testing data tested the English speaker model and the Cantonese testing data tested the Cantonese speaker model. Then we used English testing data to test the Cantonese model and Cantonese testing data to test the English model. Finally, we pooled the English and Cantonese training data for each speaker so as to train the speaker-dependent model. We refer to this as the "pooling model", and used English and Cantonese

testing data to test it. The experimental results are shown in Table 5.3 and Figure 5.4:

EER (%)	English model	Cantonese model	Pooling model
English testing	3.98	5.98	3.73
Cantonese testing	5.92	4.28	4.01

Table 5.3: Experimental results for testing language dependency in English and Cantonese.

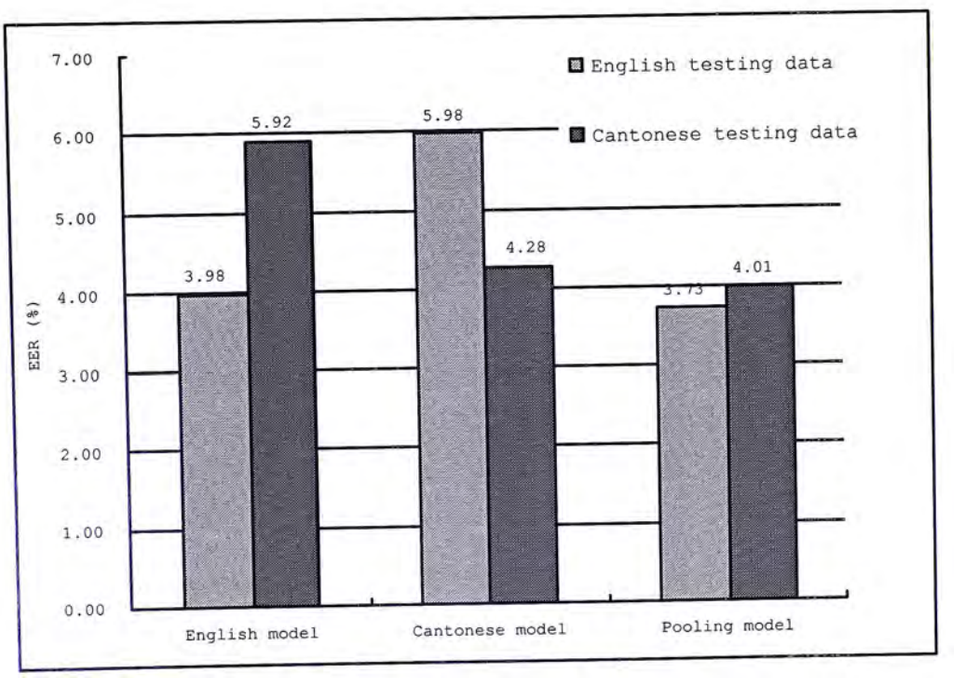

Figure 5.4: Experimental results for testing language-dependency in English and Cantonese.

The results show that using English or Cantonese to test models trained in other languages leads to SV performance degradation. For example, when using English testing data to test the speaker model trained by English data, the EER is 3.98%. If we replace the English testing data with Cantonese data, the EER increase to 5.92%. The ver-

ification performance degrades 33%. Similar results can be obtained in the case where the speaker model is trained by Cantonese data. When using Cantonese testing data to test the Cantonese model, the EER is 4.28%. If we use English testing data to test a speaker's Cantonese model, the EER rises to 5.98%. The performance declines 28%.

This shows that language dependency will affect the performance of speaker verification despite the text-independent scenario. This is because the language-dependent GMM of a speaker cannot cover the cross-language acoustic space effectively. In particular, it is not possible to effectively train a number of language-dependent acoustic features using data from other languages. The language differences related to acoustic features may be rooted in differences in speech production behavior or other factors in various languages. Hence, if the speaker's specific characteristics exist in these acoustic feature spaces, the language-dependent GMM model cannot function well when it is tested by the speech data of other languages.

This can be shown by the experimental results when speaker models are trained by pooling language-independent data. Using English testing data, the model trained by pooling data (EER=3.73%) performs better than the model trained by Cantonese data (EER=4.01%). This model even outperforms the speaker model trained by only English data (EER=3.98%). When we use Cantonese testing data to test the "pooling model", we can also observe that the GMM trained by pooling data outperforms the model trained by English. The EER decreases from 5.92% to 4.01%. Further, the pooling model beats the model trained by only Cantonese data (EER=4.28%). One possible reason for this is that more training data can cover a larger acoustic model

space. Thus, a speaker model trained by pooling data can comprise more speaker characteristics based on the larger acoustic space. The verification performance is thus improved.

Obviously, language dependency will negatively affect language-independent speaker verification performance. Therefore, in the next section, investigations will focus on language-independent speaker verification.

5.4.4 Language-Independency

In Section 5.4.3, we proved the existence of language-dependency in speaker verification. If we want to develop a perfect language-independent speaker verification system, the verification performance should not be affected by language-dependency. That is, no matter whether English or Cantonese testing data is used to test the language-independent speaker model, the speaker verification performance should be the same. In order to develop a more robust speaker model (one that has the ability to cater for language difference), we try to improve model's effectiveness by increasing the number of mixture components M with the pooling data of different languages.

In these experiments, we pool the training data in English and Cantonese from each speaker. They are in turn used to train a language-independent and text-independent speaker model. Considering that the amount of training data has risen, we can increase M step by step ($M = 256, 512, 1024$), to determine an optimal M for the language-independent TISV system. The experimental results are listed in Table 5.4.

Figure 5.5 shows that if we increase M to 512, we can get much

EER (%)	Testing data		
	English	Cantonese	Pooling
$M = 256$	3.73	4.01	3.85
$M = 512$	2.86	3.13	3.02
$M = 1024$	3.89	4.49	4.38

Table 5.4: Comparison of M for language-independent TISV systems.

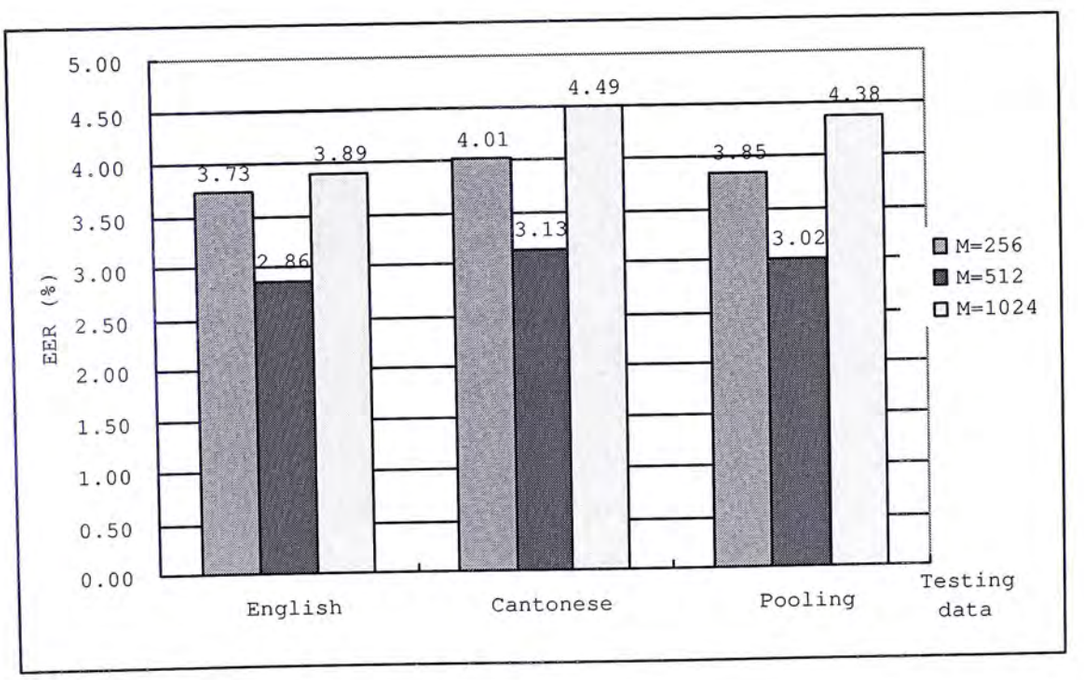

Figure 5.5: Enhanced verification performance by increasing the number of Gaussian mixture components M.

better results than $M = 256$. However, when M is 1024, the verification performance starts to drop. For example, if M increases from 256 to 512 using pooled testing data, then the EER drops from 3.85% to 3.02%. The verification performance improves by 27%. By continuously increasing M to 1024, the EER increases to 4.38%. These results are even worse than the results of $M = 256$. Similar results are obtained when using English and Cantonese testing data. We prove that $M = 512$ is the optimal number of mixture components for a speaker-dependent GMM that is trained by pooling data. We then use the results obtained when M is 512 to evaluate our system.

In order to evaluate the robustness of text-independency in our system, we compare this system to those with speaker models trained by single language data. In systems using single-language training data, M is set at 256 because there is inadequate training data to effectively train 512 mixture components in a GMM (Section 5.4.1). The results are compared in Table 5.5 and Figure 5.6.

EER (%)	Testing data		
	English	Cantonese	Pooling
English model ($M = 256$)	3.98	5.92	4.40
Cantonese model ($M = 256$)	5.98	4.28	5.25
Pooling model ($M = 256$)	3.73	4.01	3.85
Pooling model ($M = 512$)	2.86	3.13	3.02

Table 5.5: Evaluation of the language-independent TISV system.

When we use English and Cantonese testing data to test the pooling model separately, the GMM with 512 components trained by the pooling data performs best. The EER decreases from 3.73% to 2.86%

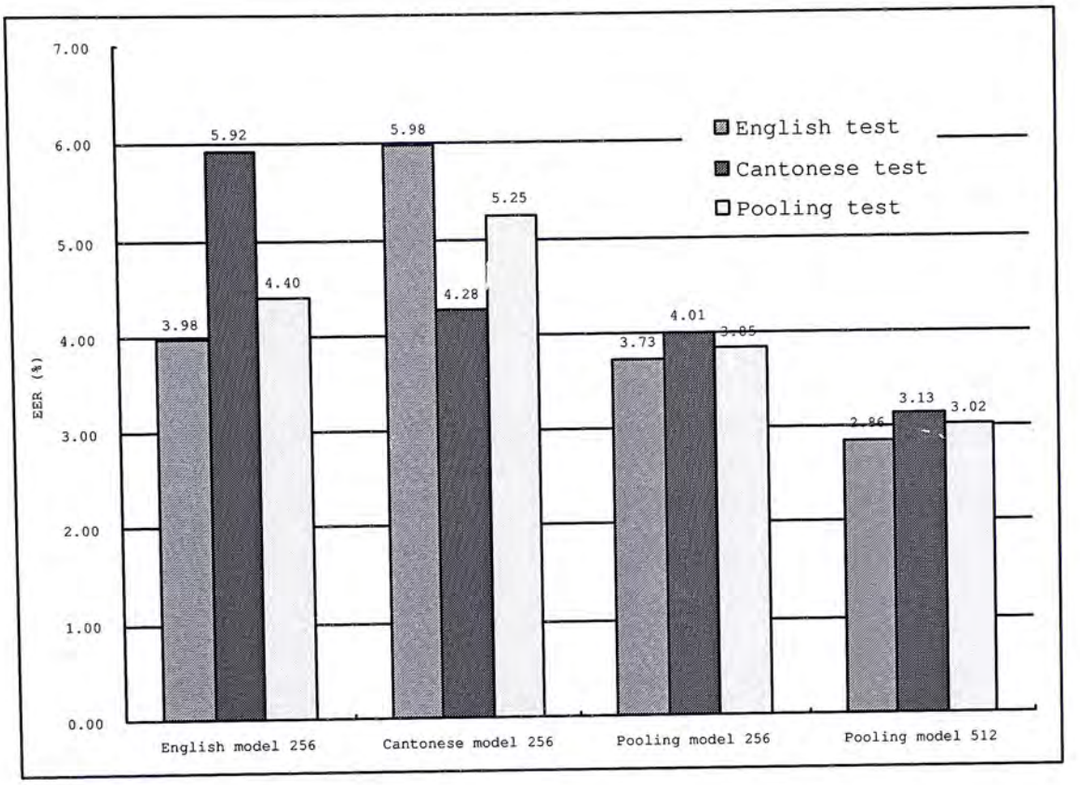

Figure 5.6: Evaluation of the language-independent TISV system.

for English testing data and from 4.01% to 3.13% for Cantonese testing data with the increment of mixture components from 256 to 512. This is due to the same reason as before; that is, using more mixture components can describe a larger acoustic space in greater detail. Therefore, speaker characteristics can be described more distinctively.

Compared with the speaker model trained by the data from a single language, the speaker model trained by pooling data performs better in the language-independent task. The experimental results, which prove this point, can be seen in Figure 5.6. The speaker's GMM with 256 mixture components, which is well-trained by pooling data, can outperform those trained by data from a single language. The best EER we can obtain is 3.85%, which is much better than the performance of the

English model (EER=4.40%) and the Cantonese model (EER=5.25%). The reason of this is that language-independent training data makes a speaker-dependent GMM more robust at dealing with differences in language.

The EERs are 4.40% and 5.25% respectively for pooling test data from the English and Cantonese models. We can observe differences in performance from both language-dependent speaker models. This is because the training data used to train the models is different for each speaker. The linguistic meaning of the training data in English and Cantonese is the same for each speaker; however, different amounts of speech data are needed to express the same linguistic meaning in the two languages. The different amount of training data for the English and Cantonese models results in diverse performance with regard to each speaker's language-dependent model.

When the number of mixture components (M) increases to 512, the SV performance improves to EER=3.02%. This is because the greater number of Gaussian mixture components in the GMM enable a more detailed description of a speaker's distinctive characteristics. This in turn enables a more detailed classification of speakers' characteristics, which leads to improved verification performance.

The data from our different language tests on our pooling data language-independent SV system shows that the EERs of verification are 2.86% and 3.13% for English and Cantonese respectively. Differences in verification performance are slight. Ideally, in a perfect language-independent SV system, this should not occur. However, speaker models trained by pooling training data in different languages is still a direct and effective way to solve the language independency

problem for SV systems.

5.5 Chapter Summary

In this chapter, we have described how to develop a language-independent TISV system using GMM-based modeling techniques. First, we developed experiments to find the optimal number of Gaussian mixture components (M) that can be trained by the available training data. Second, we applied cohort normalization techniques to the system. This technique led to significant improvements in verification performance. Specifically, the EERs decreased as much as five times. Our experiments for testing language-dependency proved that a language-dependent effect exists in Cantonese and English TISV systems. The language-dependent effect refers to the differences in verification performance when data from different languages are used to test language-dependent speaker models. We have been able to prove that pooling data is an effective solution to problems associated with language-independent TISV. Our proof is that when the speaker models trained by pooled data from different languages are tested by pooled test data, the EERs decrease from 4.40% and 5.25% for English and Cantonese speaker models respectively to 3.86% for both. Pooled data from several languages along with an increment of mixture components led to much better verification performance. Finally, we developed a bilingual and TISV system with a performance of EER = 3.02% by increasing the number of mixture components. Compared with a result of EER=3.86% when the number of mixture components was not increased, the verification performance improved by 22%.

□ **End of chapter.**

Chapter 6

Conclusions and Future Work

In this chapter, we will first summarize our works in Section 6.1. The summary is described from the points of view of feature extraction, speaker modeling, cohort normalization and language-dependency of a speaker verification system. Based on our current research, some valuable future works are proposed in Section 6.2.

6.1 Summary

In this thesis, we have presented a methodology for using the statistical technique to develop TDSV and bilingual TISV systems. Specifically, HMM and GMM techniques are employed in our work. This summary focuses on four points: feature comparison, speaker modeling, cohort normalization, and language-dependency effect.

6.1.1 Feature Comparison

In our TDSV baseline systems, we learnt how to set up the basic components of a speaker verification system. Signal representation analysis was briefly introduced based on linear predictive analysis and filter bank analysis. Feature vectors in cepstral domain are employed to describe speaker voice characteristics. Experiments on the YOHO corpus showed that LPCC outperform MFCC using our feature extraction setup.

6.1.2 HMM Modeling

In our investigation, an HMM-based TDSV system is developed based on the YOHO corpus. In an HMM-based TDSV system, the speaker-dependent model consists of a set of left-to-right continuous density HMMs. These HMMs are trained with respect to the possible acoustic units. Subwords are the acoustic unit used in our work. Furthermore, we use two kinds of HMMs to model the non-speech segments of utterances; that is,. the silence model and the short pause model. Speaker-dependent HMMs can describe the speaker-specific characteristics according to the acoustic units. The use of HMM in a TDSV system have been proved to be very effective in our experiments.

6.1.3 GMM Modeling

GMM can be deemed as a single-state HMM. It can provide a probabilistic model of the underlying characteristics of a person's voice. However, unlike HMM, it does not impose any Markovian constraints between the sound classes. It is fair to say that GMM discards the

phonetic information of the speakers' utterances, but only describes speaker-specific characteristics. GMM-based speaker verification is investigated with the YOHO corpus and the CUBS corpus. With regard to the YOHO corpus, GMM shows its effectiveness on model speaker characteristics in a small vocabulary constraint scenario. Therefore, we extend the scalability of GMM modeling technique to the language- and text-independent scenario on the CUBS corpus.

6.1.4 Cohort Normalization

Because of variations in a speaker's voice characteristics, one can never repeat an utterance precisely. These variations can also arise from background noises or transmission effects, etc. In order to compensate for these variations, we adopt the cohort normalization technique. This technique uses the relative scores between the claimed speaker and a set of his/her closest speakers to replace the absolute likelihood score in decision. Experimental results show that if the cohort set is well-selected, cohort normalization can greatly improve the verification performance.

6.1.5 Language Dependency

Another contribution of this thesis is that we investigated the language dependency of the TISV system. We develop a bilingual text-independent corpus, named CUBS, for this task. Our experiments prove the existence of language-dependency for English and Cantonese. Through a number of experiments, we observed that the TISV system's performance was affected by the language-dependency of English and Cantonese. This is due to the existence of the language-dependent

speaker characteristics of different languages. These characteristics are useful for distinguishing speakers. In our investigations, we pooled speakers' training data in different languages and trained their speaker-dependent GMM. Through increasing the number of Gaussian mixture components, we developed a bilingual TISV system.

6.2 Future Work

A speaker verification system includes many technologies, such as speech signal processing, pattern recognition and hypothesis testing, etc. A confidential speaker verification system also has great market value. Hence, in order to improve the performance of our speaker verification system, we have listed a number of possible extensions from this work.

6.2.1 Feature Parameters

Previous work has investigated numerous kinds of signal processing techniques and feature parameters for speaker verification systems. Our comparison of LPCC and MFCC speaker verification systems is not sufficient. There is still no perfect setup for feature extraction in different speaker verification systems. Future work will need to solve problems associated with this issue.

6.2.2 Model Quality

6.2.2.1 Variance Flooring

In order to make the enrollment process convenient to users of a speaker verification system, only very few enrollment data can be made available. One problem of using small training data sets is the risk of

over-training in ML estimation in the statistical speaker verification method. As hinted by the name, over training are those parameters of the client model that are over-fitted to the particular training data. In particular, variance parameters are susceptible to over-fitting. This means that variance estimated from only a few data points can be very small and might not be representative of the underlying distribution of the data source.

In order to prevent over-training, the approach we proposed is to use a speaker-independent variance to replace the over-fitting variance. For example, we can set a floor for the variance value, with the variance lower than the floor (over-fitting), then the variance can be copied from a well-trained gender-dependent, multi-speaker, non-client model. Using this variance flooring technique can ensure a more reliable likelihood score.

6.2.2.2 Silence Detection

It is known that the non-speech segments of utterances do not provide information about the speaker's voice characteristics. Hence, to use these data to train speaker models is not useful. In our GMM based TISV system, we tried using a simple method to carry out silence detection and remove the non-speech part in training and testing. However, the speech and non-speech segments can not be easily separated. Doing so leads to time alignment errors and the likelihood score of the observation and the model decreases. Therefore, an effective silence detection technique will be helpful to improve the speaker verification performance.

6.2.3 Conversational Speaker Verification

As mentioned earlier, the variations caused by the speaker's voice characteristics and background noise or transmission channel are the most significant factors affecting a verification system's performance. Other than channel compensation and normalization techniques, it could be useful to incorporate knowledge-based decisions into the verification system, so as to obtain higher security.

This approach is based on a text-independent speaker verification system. First, a user's profile and data - such as "favorite color", "staff ID"and "birthday", etc. - is stored in the database . When combined with a dialog system, the questions addressed to the user can be randomly selected, following a pre-defined sequence or a specific logical path. With this approach, user verification relies on both acoustic recognition and the content of the answers to questions. Hence, a combined decision according to "what you are" and "what you know" can be obtained, which is beyond the traditional speaker verification that relies on "what you are".

With these improvements, our language independent speaker verification system can be more flexible in carrying out authentication, especially if users are not constrained by the limitations of language limitation.

□ **End of chapter.**

Bibliography

[1] W. H. Abdulla and N. K. Kasabov. The concepts of hidden markov model in speech recognition. *Technical Report of Knowledge Engineering Lab Information Science Department University of Otago*, Sep 1999.

[2] R. Auckenthaler, M. J. Carey, and J. S. D. Mason. Language dependency in text-independent speaker verification, 2000.

[3] F. Bimbot, M. Blomberg, and L. Boves. An overview of the cave project research activities in speaker verification. *Speech Communication*, 31(2-3):158–180, 2000.

[4] J. P. Campbell. *Features and Measures for Speaker Recognition.* PhD thesis, Oklahoma State University, 1992.

[5] J. P. Campbell. Testing with the yoho cd-rom voice verification corpus. *ICASSP-95*, pages 341–344, 1995.

[6] J. P. Campbell. Speaker recognition: A tutorial. In *Proceedings of the IEEE*, volume 85, pages 1437–1462, Sep 1997.

[7] C. W. Che, Q. guang Lin, and D.-S. Yuk. An hmm approach to text-prompted speaker verification. *IEEE Conf. on Acoustics, Speech, and Signal Processing*, 1996.

[8] K. Chen. Towards better making a decision in speaker verification. *Pattern Recognition*, 36(2):53–70, 2003.

[9] Y. Cheng. Speaker verification over the telephone. Master's thesis, The Chinese University of Hongkong, 1999.

[10] A. Cohen and Y. Zigel. On feature selection for speaker verification. *Proceedings of COST 275 workshop on The Advent of Biometrics on the Internet*, pages 89–92, Nov 2002.

[11] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, Signal Processing*, ASSP-28(4), August 1980.

[12] J. R. Deller, J. G. Proakis, and J. H. Hansen. *Discrete-time Processing of Speech Signals*, chapter 6, pages 353–397. New York: Macmillan Pub. Co. ; Toronto : Maxwell Maxmillan Canada,, 1993.

[13] G. Doddington and M. A. Przybocki. The nist speaker recognition evaluation - overview, methodology, systems, results, perspective. *Speech Communication*, 31:225–254, 2000.

[14] J. P. et. al. Signal modeling techniques in speech recognition. *Proceedings of the 1993 IEEE Automatic Speech Recognition and Understanding Workshop*, 81(9):1215–1247, 1993.

[15] H. Hermanskey and N. Morgan. Rasta processing of speech. *IEEE Trans. on Speech and Audio Processing*, 2(4):578–589, 1994.

[16] Higgins, L. Bhaler, and J. Porter. Voice identificationi using nearest neighbor distance. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1:269–272, 1990.

[17] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development.* Prentice-Hall,Inc, 2001.

[18] D. Lee. Fundamentals of speech recognition: Lecture notes. The Chinese Unversity of Hongkong, 2001.

[19] Q. Li, B.-H. Juang, Q. Zhou, and C.-H. Lee. Automatic verbal information verification for user authentication. *IEEE Transactions on Speech and Audio Processing*, 8(5):585–596, Sep 2000.

[20] Q. Li, B.-H. Juang, Q. Zhou, and C.-H. Lee. Recent advancements in automatic speaker authentication. *IEEE Robotics and Automation Magazine*, pages 24–34, March 99.

[21] C.-S. Liu, H.-C. Wang, and C.-H. Lee. Speaker verification using normalized log-likelihood score. *IEEE Transactions on Speech and Audio Processing*, 4(1):56–60, Jan 1996.

[22] L. Liu, J. He, and G. Palm. Signal modeling for speaker identification. *IEEE Transctions on Speech and Audio Processing*, 1996.

[23] S. H. Maes, J. Navratil, and U. V. Chaudhari. Conversational speech biometrics. *E-Commerce Agents*, pages 166–179, 2001.

[24] J. Makhoul. Spectral analysis of speech by prediction. *IEEE Trans. Acoustics Speech and Signal Processing*, 21(3):140–148, 1973.

[25] J. Makhoul. Linear prediction: A tutorial review. *IEEE Conf. on Acoustics, Speech, and Signal Processing*, pages 561–580, 1975.

[26] T. Matsui and F. Sadaoki. Comparison of text-independent speaker recognition method using vq-distortion and discrete/continous hmms. *IEEE Conf. on Acoustics, Speech, and Signal Processing*, pages 157–160, 1992.

[27] D. O'Shaughnessy. Speaker recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Oct 1986.

[28] K. Paliwal. *Advances in Speech, Hearing, and Language Processing.*, volume 1. London ; Greenwich, Conn, 1990.

[29] M. Pandit and J. Kittler. Feature selection for a dtw based speaker verification system. *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pages 769–772, 1998.

[30] P. G. POP and E. LUPU. Speaker verification with vector quantisation. *Internet Workshop: TRENDS AND RECENT ACHIEVEMENTS IN INFORMATION TECHNOLOGY*, May 2002.

[31] X. Qing and K. Chen. On use of gmm for multilingual speaker verification: An empirical study. *Proceedings of International Symposium on Chinese Spoken Language Processing*, pages 263–266, 2000.

[32] L. Rabiner. A tutorial on hidden markov models and selected application in speech recognition. *Proceedings of IEEE*, 77:257–286, 1989.

[33] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition.* Prentice-Hall, Englewoods Cliffs, 1993.

[34] D. A. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, 17(1-2):91–108, 1995.

[35] A. Rosenberg and F. Soong. Recent research in automatic speaker recognition. *Advances in Speech Signal Processing*, pages 701–738, 1992.

[36] F. Sadaoki. Cepstral analysis technique for automatic speaker verification. In *Acoustics, Speech, and Signal Processing*, volume 29, pages 254–272, Apr 1981.

[37] F. Sadaoki. Research on individuality features on speech waves and automatic speaker recognition techniques. *Speech Communication*, 5:183–197, 1986.

[38] F. Sadaoki. *Digital Speech Processing, Synthesis, and Recognition*, chapter 9, page 349. New York: Marcel Dekker, 2nd edition, 2001.

[39] F. Sadaoki, C.-H. Lee, F. Soong, and K. K. Paliwal. An overview of speaker recognition technology. *Automatic Speech And Speaker Recognition*, pages 31–56, 1996.

[40] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[41] N. Samir. *Biometrics: Identity Verification in a Networked World.* New York John Wiley & Sons, 2002.

[42] B. Schoner. *State Reconstruction for Determining Predictability in Driven Nonlinear Acoustical Systems.* Ph.d, Massachusetts Institute of Technology, Month 1996.

[43] A. R. Setlur, R. A. Sukkar, and M. B. Gandhi. Speaker verification using mixture likelihood profiles extracted from speaker independent hidden markov models. *IEEE Conf. on Acoustics, Speech, and Signal Processing*, 1996.

[44] F. Soong, A. Rosenberg, and L. Rabiner. A vector quatization approach to speaker recognition. *AT&T Tech.J*, 66(2):14–26, 1987.

[45] F. Soong and A. Rossenberg. On the use of instaneous and transitional spectral information in speaker recognition. *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1986.

[46] T.Matsui and F. Sadaoki. Text-independent speaker recognition using vocal track and pitch information. *Proc. of the International Conference on Spoken Language Processing*, pages 137–140, 1990.

[47] R. Vergin, D. O'Shaughnessy, and A. Farhat. Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition. *IEEE Trans. on Speech and Audio Processing*, 7(5):525–532, 1999.

[48] R. Vergin, D. O'Shaughnessy, and V. Gupta. Compensated mel-frequency cepstrum coefficients. *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May 1996.

[49] J. Woodward. Biometrics: Privacy's foe or privacy's friend? In *Proceedings of the IEEE*, volume 85, pages 1480–1492, Sep 1997.

[50] S. Young and D. Kershaw. The htk book (for htk v3.0), Sep 2001.

[51] D. Zhang. *Automated Biometrics Technologies and Systems.* Kluwer Academic Publishers, 2000.