Towards high performance phonotactic feature for spoken language recognition

Tong, Rong

2012

Tong, R. (2012). Towards high performance phonotactic feature for spoken language recognition. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/50585>

<https://doi.org/10.32657/10356/50585>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**TOWARDS HIGH PERFORMANCE
PHONOTACTIC FEATURE FOR SPOKEN
LANGUAGE RECOGNITION**

TONG RONG

SCHOOL OF COMPUTER ENGINEERING

2012

TOWARDS HIGH PERFORMANCE PHONOTACTIC FEATURE FOR SPOKEN LANGUAGE RECOGNITION

TONG RONG

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2012

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Li Haizhou, department head of Human Language Technology, Institute for Infocomm Research (I²R). I am grateful to him for introducing me to the area of speech and language processing, and inspiring me throughout my candidature.

I am deeply indebted to my supervisor, Dr. Chng Eng Siong, School of Computer Engineering, Nanyang Technological University (NTU). I'm grateful for his patience in correcting and editing of my manuscript. As a part time student, I often struggle between study and work load. Dr. Chng has always supported me and give me much freedom on my schedule.

I would also like to thank my mentor, Dr. Ma Bin, group leader of speech processing, Institute for Infocomm Research, for his mentorship, helpful discussion, valuable suggestions and encouragement in the process of the research. This piece of work will never be accomplished without his support.

My sincere thanks to all my colleagues in Human Language Interface Department, Institute for Infocomm Research (I²R), thank them for helping me in various ways. It is a great experience to work with these excellent researchers.

I would like to express my special thanks to my friends in Nanyang Technological University, especially Mr. Wang Lei and Dr. Xiao Xiong, who gave me tremendous help in graduate school related matters. I would like to thank Mr. Jonathan William Dennis for the outstanding job he did on proof reading my thesis.

Finally but most importantly, I would like to thank my family, my husband for encouraging and believing that I could get through, my two sweet daughters for being cheering me up all the way. I have always felt their presence by my side whenever I meet problems. My work would not have been possible without their love, concern, continuous support and encouragement.

Contents

Acknowledgments	1
Abstract	7
List of Publications	9
List of Figures	12
List of Tables	14
List of Abbreviations	16
1 Introduction	18
1.1 Discriminative cues for language recognition	19
1.2 Problem statement and contribution	21
1.3 Discriminative phonotactic feature extraction	23
1.3.1 Target oriented phonotactic feature extraction	23
1.3.2 Target-aware language models	24
1.3.3 Automatic relevance feedback for short utterances	24
1.3.4 Feature selection to reduce redundant phonotactic information . .	25
1.4 Organization of the thesis	26
2 Related Works	28
2.1 General process of automatic languages recognition	28
2.2 Performance measures	30
2.2.1 Performance measures for language identification	30
2.2.2 Performance measures for language verification	31
2.2.3 NIST language recognition evaluation	32
2.3 Review of automatic language recognition techniques	35
2.3.1 Phonotactic feature based methods	35

2.3.2	Acoustic feature based methods	38
2.3.3	Prosodic feature based methods	40
2.4	Summary	42
3	Phonotactic Feature Based Language Recognition	44
3.1	Techniques for phonotactic feature based language recognition	45
3.1.1	Phone recognizer as front-end	45
3.2	PPR-VSM language recognition system	50
3.2.1	VSM back-end	51
3.2.2	Recognition	52
3.2.3	PPR-LM versus PPR-VSM	53
3.3	Summary	54
4	Target-oriented Front End for Automatic Language Recognition	55
4.1	Objective	55
4.2	Target-oriented phone tokenizer	56
4.2.1	Phone selection strategy	58
4.2.2	Tokenizer selection	60
4.3	Phone Recognizers	63
4.3.1	Language specific phone recognizers	63
4.3.2	Universal phone recognizers	65
4.4	Experiments	66
4.4.1	Phone selection	67
4.4.2	TOPT selection	71
4.4.3	Bigram vs. trigram phonotactic statistics	73
4.4.4	Language specific phone recognizer vs. universal phone recognizer	75
4.4.5	1-best vs. lattice phonotactic statistics	78
4.5	Summary	79

5	Target-aware Language Models for Language Recognition	81
5.1	Introduction	81
5.2	Target-aware language models construction	83
5.3	Discriminative ability of phones	84
5.4	Target-aware language models	84
5.5	Experiments on TALM	86
5.5.1	Language recognition with TALM	86
5.5.2	Higher order language models	88
5.5.3	Discriminative ability of phones from UPR-Merge	89
5.5.4	TOPT vs TALM	90
5.6	Summary	91
6	Language Recognition with Relevance Feedback	93
6.1	Introduction	93
6.2	Relevance feedback for language recognition	95
6.2.1	Query expansion with relevant utterances	96
6.2.2	Query expansion with relevant clusters	97
6.2.3	Query expansion with relevant and irrelevant utterances	99
6.3	Experiments on query expansion	100
6.3.1	Baseline system	101
6.3.2	Query expansion by relevant utterances	101
6.3.3	Query expansion by relevant clusters	102
6.3.4	Query expansion by relevant and non-relevant utterances	103
6.3.5	Conclusion	104
6.4	Summary	105
7	Phonotactic Feature Selection for Language Recognition	106
7.1	Introduction	106
7.2	Phonotactic feature selection	107
7.2.1	Previous works	108
7.2.2	Feature selection with separation margin (SM) and Chi-squared measure (CM) criteria	110

7.2.3	Efficient feature selection with relative mean difference (MD) . . .	112
7.3	Experiments	114
7.3.1	Experiment setup	114
7.3.2	Comparison of different feature selection criteria	114
7.3.3	Validation on the feature selection with relative mean difference (MD)	116
7.3.4	Combining two criteria	117
7.3.5	Incorporating higher order n -gram features in language recognition	118
7.4	Summary	121
8	Conclusions	122
8.1	Contributions	123
8.1.1	Target-oriented front-end	123
8.1.2	Target-aware language models	123
8.1.3	Language recognition with relevance feedback	124
8.1.4	Phonotactic feature selection	124
8.2	Conclusions and future works	125
	Appendix	127
A	Corpus	127
A.1	Language corpus for training	127
A.2	Language corpus for testing	129
	References	131

Abstract

With the demands of globalization, multilingual speech is increasingly common in conversational telephone speech, broadcast news and internet podcasts. Therefore, automatic spoken language recognition has become an important technology in multilingual speech related applications. For example, automatic spoken language recognition has been used as a preprocessing component for spoken language translation, multilingual speech recognition and spoken document retrieval.

Both humans and machines rely on certain informative cues to differentiate one language from another. Inspired by the findings in the discriminative cues for human language recognition, most of the automatic language recognition systems rely on the following three features: acoustic, prosodic and phonotactic. Acoustic features capture spectral characteristics and can be obtained from short-term speech signals. Prosodic features such as tone, intonation, prominence and rhythm can be derived from energy measurements, pitch contour, rate of change. Phonotactic features capture the statistics of lexical constraints and phonotactic patterns.

Phonotactic features can be generated from a tokenization front end which converts speech signals into sequences of sound patterns. This thesis focuses on the study of effective phonotactic feature extraction methods for high performance automatic language recognition. Specifically, the main contributions of this thesis are:

A novel target-oriented method is proposed to construct parallel phone recognizers for robust phonotactic feature extraction. A subset of the most discriminative phones from an existing phone recognizer is selected to form a target-oriented phone tokenizer (TOPT). The TOPT phone tokenizers, one for each of the target languages, are constructed from an existing phone recognizer without requiring additional transcribed training data.

A target-aware language models (TALM) method is proposed to generate phone tokenizers by constructing a set of phone language models, each dedicated to a target language. In the front-end decoding process with TALM, all the phone models of the original phone recognizer are used, and they are constrained by target-aware language models. Each target-aware language model emphasize on the discriminative ability of phones for a specific target language.

An automatic relevance feedback technique is proposed to incorporate more language information in language recognition with short utterances. The idea is to augment the short input utterance with relevant utterances from the reference corpus. In this way, the short utterances are augmented with richer information and better language recognition accuracy can be achieved.

A feature selection method is proposed to reduce redundant phonotactic information to make the language recognition system more efficient. The dimensional reduction is achieved by measuring the importance of features using two different criteria: contribution to SVM separation margin and Chi-squared value.

List of Publications

Journal (2)

- (i) **R. Tong**, B. Ma, H. Li and E. S. Chng, “A target-oriented phonotactic front-end for spoken language recognition,” *IEEE Transactions on Audio, Speech and Language Processing*, Volume 17, issue 7, pp. 1335-1347, September, 2009. **Google: Cited by 2.**
- (ii) B. Ma, H. Li and **R. Tong**, “Spoken language recognition using ensemble classifiers,” *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, issue 7, pp. 2053-2062, September 2007. **Google: Cited by 10.**

Conference (17)

- (i) **R. Tong**, B. Ma, H. Li and E. S. Chng, “Target-aware lattice rescoring for dialect recognition,” *Interspeech 2011*, August 2011, Italy.
- (ii) **R. Tong**, B. Ma, H. Li and E. S. Chng, “Selecting phonotactic features for language recognition,” *Interspeech 2010*, September 2010, Japan.
- (iii) **R. Tong**, B. Ma, H. Li and E. S. Chng, “Target-aware language models for spoken language recognition,” *Interspeech 2009*, 6 - 10 September 2009, UK.
- (iv) **R. Tong**, B. Ma, H. Li and E. S. Chng, “Target-oriented phone selection from universal phone set for spoken language recognition,” *Interspeech 2008*, 22 - 26 September 2008, Australia. **Google: Cited by 2.**

- (v) **R. Tong**, B. Ma, H. Li and E. S. Chng, “Target-oriented phone tokenizers for spoken language recognition,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008)*, pp. 4221-4224, Las Vegas, March 2008, USA. **Google: Cited by 7.**
- (vi) **R. Tong**, B. Ma, D. Zhu, H. Li and E. S. Chng, “Spoken language recognition with relevance feedback,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007)*, pp. 861-864, Hawaii, April 2007, USA.
- (vii) **R. Tong**, B. Ma, D. Zhu, H. Li and E. S. Chng, “Integrating acoustic, prosodic and phonotactic features for spoken language identification,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 14-19, 2006 Toulous, France. **Google: Cited by 33**
- (viii) **R. Tong**, B. Ma, K. Lee, C. You, D. Zhou, T. Kinnunen, H. Sun, M. Dong, E. S. Chng and H. Li, “Fusion of acoustic and tokenization features for speaker recognition,” *5th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, December 2006, Singapore. **Google: Cited by 8.**
- (ix) Kumar, C.P. Santhosh, H. Li and **R. Tong**, P. Matejka, L. Burget and J. Cernocky, “Tuning phone decoders for language identification,” *ICASSP 2010*, pp. 5010-5013, March, 2010, USA.
- (x) C. C. Leung, **R. Tong**, B. Ma and H. Li, “A lattice-based phonotatic language recognition system with CMLLR adaptation and its implementation issues,” *IALP 2009*, pp. 285-288, December 2009, Singapore.
- (xi) H. Li, B. Ma, K.A. Lee, H. Sun, D. Zhu, K. C. Sim, C. You, **R. Tong**, I. Karkkainen, C. Huang, V. Pervouchine, W. Guo, Y. Li, L. Dai, M. Nosratighods, T. Tharmarajah, J. Epps, E. Ambikairajah, E.S. Chng, T. Schultz and Q. Jin, “The I4U system in NIST 2008 speaker recognition evaluation,” *ICASSP 2009*, pp. 4201-4204, April 2009, Taipei, Taiwan.

- (xii) H. Li, B. Ma, K.-A. Lee, K.-C. Sim, H. Sun, **R. Tong**, D. Zhu, and C. You, “NIST 2007 language recognition evaluation: from the perspective of IIR,” *22nd Pacific Asia Conference on Language, Information and Computation (PACLIC)*, Cebu City, 20-22 November, 2008, Philippines.
- (xiii) B. Ma, **R. Tong** and H. Li, “Discriminative vector for spoken language recognition,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007)*, pp. 1101-1104, Hawaii, April 2007, USA. **Google: Cited by 6.**
- (xiv) B. Ma, D. Zhu and **R. Tong**, “Tone features based on pitch flux for Chinese dialect identification,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1029-1032, May 14-19, 2006 Toulouse, France. **Google: Cited by 12**
- (xv) H. Li, B. Ma, and **R. Tong**, “Vector-based spoken language recognition using output coding,” *Interspeech 2006*, September 17-21, 2006, Pittsburgh, USA. **Google: Cited by 5.**
- (xvi) K. Lee, H. Sun, **R. Tong**, B. Ma, M. Dong, C. You, D. Zhu, C. E. Koh, L. Wang, T. Kinnunen, E. S. Chng, and H. Li, “The IIR submission to CSLP 2006 speaker recognition evaluation,” *5th International Symposium on Chinese Spoken Language Processing, ISCSLP*, pp. 861-870, December 2006, Singapore.
- (xvii) J. Li, S. Yaman, C.H. Lee, B. Ma, **R. Tong**, D. Zhu, and H. Li, “Language recognition based on score distribution feature vectors and discriminative classifier fusion,” *Speaker and Language Recognition Workshop, 2006*, IEEE Odyssey 2006: June 2006, pp. 1-5, San Juan, Puerto Rico. **Google: Cited by 5.**

List of Figures

1.1	Five levels of features for language recognition	20
2.1	General automatic language recognition process	29
2.2	DET curve for language recognition	34
3.1	PR-LM system with M target languages	46
3.2	PPR-LM system with F phone recognizers and M target languages . . .	48
3.3	Phone lattice of word ‘seven’ generated by a English phone recognizer . .	50
3.4	PPR-VSM language recognition system with F phone recognizers	51
4.1	Target-oriented phone tokenizer	57
4.2	Discriminative index of English phones for the target language Mandarin based on separation margin (SM) and mutual information (MI) criteria .	68
4.3	Language recognition using 12 English TOPTs with different numbers of phones in each of the TOPTs on LRE03 30-second trials	69
4.4	The number of times each phone in the English phone recognizer is selected in the 12 TOPTs	70
4.5	Merit, discriminative ability and distinctiveness of TOPTs derived from the English phone recognizer	71
4.6	Language recognition performance (EER%) as a function of the number of TOPTs	73
4.7	Performance comparison of bigram vs trigram phonotactic features on the LRE03 30-second trials	74
5.1	Target-aware language models: TALM	83

5.2	Distribution of the top 50 phones in UPR-Merge-TALM for English language recognition	90
5.3	EER (%) of different number of phones in UPR-Merge-TOPT and UPR-Merge-TOPT-TALM on LRE03 30-second test	91
6.1	Speech query expansion with relevant utterances	97
6.2	Speech query expansion with relevant clusters	98
6.3	Speech query expansion by relevant and irrelevant utterances	100
7.1	Features ranked by contribution to separation margin (merit values in $\log_e()$ scale	112
7.2	EER (%) with different numbers of features selected by different selection criteria on the LRE07 30-second test set	115
7.3	Mean difference of ranked features derived by separation margin criterion on the LRE07 30-second set	117
7.4	EER (%) with different number of features selected by separation margin criterion on LRE07 30-second set	118
7.5	Mean difference of ranked features derived by the Chi-squared criterion on the LRE07 30-second set	119
7.6	EER (%) with different number of features selected by the Chi-squared criterion on the LRE07 30-second set	120
7.7	EER (%) with different feature selection criteria	121

List of Tables

4.1	Phone recognizers used in the thesis	64
4.2	Language recognition performances (EER%) of the original English phone recognizer (44 phones) and an English phone tokenizer with only 36 selected phones	70
4.3	Performance comparison between Top5 and Bottom5	72
4.4	Language recognition performance with UPR and language specific phone recognizer as front-end tokenizers (EER%)	76
4.5	Language recognition performance (EER%) with TOPTs derived from UPR and from language specific phone recognizers as front-end tokenizers. In brackets are the relative EER reductions of the TOPT front-ends over their original phone recognizers	76
4.6	Language recognition performance with UPR and their derived TOPTs as front-end tokenizers (EER%)	77
4.7	Language recognition performance with TOPTs derived from UPR-Merge using lattice and one best phonotactic statistics (EER%)	79
5.1	EER (%) of language recognition systems on 30-second test trials, with English phone inventory	87
5.2	EER (%) of language recognition systems on 30-second test trials with merged phone inventory	87
5.3	EER (%) of language recognition systems on 30-second test trials with merged phone inventory using different levels of n -gram language models	89
6.1	Baseline system equal error rates (EER%) on NIST 1996, 2003, 2005 LRE tasks with Hungarian phone recognizer.	101

6.2	Equal error rates (EER%) of query expansion with relevant utterances on NIST 1996, 2003, 2005 LRE tasks with Hungarian phone recognizer. . . .	102
6.3	Equal error rates (EER%) of query expansion with relevant clusters on NIST 1996, 2003, 2005 LRE tasks with Hungarian phone recognizer. . . .	103
6.4	Equal error rates (EER%) of query expansion with relevant and irrelevant utterances on NIST 1996, 2003, 2005 LRE tasks with Hungarian phone recognizer.	104
6.5	Equal error rates (EER%) for 3-second test queries of NIST 1996, 2003, 2005 LRE tasks with PPR1 phone recognizers. The percentages in the brackets are the relative improvement compared to baseline.	105
7.1	Contingency table for a feature and a target language	111
7.2	EER (%) with different order of n -gram and the number of selected features	119
A.1	Languages/dialects in the development data	128
A.2	Target languages/dialects and number of trials in the NIST LREs	129

List of Abbreviation

ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
BUT	Brno University of Technology
CM	Chi-squared Measure
DCF	Detection Cost Function
DET	Detection Error Tradeoff
EER	Equal Error Rate
EM	Expectation Maximization
FA	False Alarm
FAR	False Alarm Rate
MS	Miss
MSR	Miss Rate
GLDS	Generalized Linear Discriminant Sequence
GMM	Gaussian Mixture Model
GSV	Gaussian Super Vector
HMM	Hidden Markov Model
IDR	Identification Rate
IIR	Institute for Infocomm Research
LDA	Linear Discriminant Analysis
LDC	Linguistic Data Consortium
LFA	Latent Factor Analysis
LM	Language Model
LMSI	Laboratoire d'Informatique pour la Mecanique et les Sciences de l'Ingenieur
LPCC	Linear Prediction Cepstrum Coefficient
LRE	Language Recognition Evaluation
LVCSR	Large Vocabulary Continuous Speech Recognition
MD	Mean Difference
MFCC	Mel-Frequency Cepstral Coefficient
MI	Mutual Information
MLLR	Maximum Likelihood Linear Regression
MMI	Maximum Mutual Information

NAP	Nuisance Attribute Projection
NIST	National Institute of Standards and Technology
OGI-TS	Oregon Graduate Institute Multi-Language Telephone Speech Corpus
OOS	Out Of Set
PCA	Principal Component Analysis
PLP	Perceptual Linear Predictive
PR	Phone Recognizer
PPR	Parallel Phone Recognizers
PPR-LM	Parallel Phone Recognition followed by Language Modeling
PR-LM	Phone Recognition followed by Language Modelling
PR-VSM	Phone Recognition followed by Vector Space Modeling
SDC	Shifted Delta Cepstra
SM	Separation Margin
SVM	Support Vector Machine
TALM	Target-aware Language Model
TFIDF	Term Frequency Inverse Document Frequency
TRAP	TempoRAi Patterns
TOPT	Target-Oriented Phone Tokenizer
UBM	Universal Background Model

Chapter 1

Introduction

Automatic spoken language recognition seeks to determine the language identity from a sample of speech. Generally, there are two types of tasks for an automatic language recognition system: language identification and language verification. A language identification system answers the question of what is the language identity of a given speech segment. For the language identification task, it is assumed that the input speech is from a fixed set of known languages, thus the language identification task is sometimes also referred to as a closed-set identification task. Alternatively, in the language verification task, the objective is to accept or reject the hypothesis that a speech segment is spoken in a certain language. There are two types of verification tasks: closed-set and open-set verification. In the closed-set language verification task, all the test segments are spoken in languages that are known to the system. In the open-set language verification task, not all of the test utterances' language identities are known to the system.

Although humans can recognize well the speech in their familiar languages, their ability is discounted when they encounter unfamiliar languages or when they need to make decisions on a large pool of languages. Recognizing a new language is challenging to us, as identifying a language among others requires enough exposure of these languages. It is noted that an automatic language recognition system is able to recognize a new target language in a relatively short time given sufficient training data from those languages.

As multilingual speech is increasingly common in applications involving conversational telephone speech, broadcast news and the internet, automatic spoken language recognition has become an essential technology. In a multilingual speech application, a language recognition system serves as a preprocessing component to identify the language of the

input speech segments. For example, in a speech translation [3] system, an unknown language speech segment can first be identified by a language recognition component prior to being processed by its corresponding language specific machine translation component. For the case of a multilingual speech recognition [4] system, the language identity is identified so that the appropriate speech recognizer is applied for recognition. For a spoken document retrieval [5] system, the language identity is required for the use of specific lexical and grammar information by the document retrieval component.

We know that language identity is an implicit information that is carried by the speech, the realization of which is subject to many factors. Such factors can be broadly grouped into two categories: subjective and objective distortions. Subjective distortion includes variations introduced by the speaker. For example, the educational background of the speaker affects the word or syntax used. The age and gender of the speaker determine the pitch and energy level of the speech signal. The emotion of the speaker changes the intonation factor, and pronunciation differs between native and non-native speaker. Objective distortion includes distortions introduced by environmental factors. For example, different transmission channels or sample rates affect the spectral information in the speech. The background noise and the distance of the microphone to the speaker also affect the speech signal.

As the performance of an automatic language recognition system is greatly influenced by the above-described distortions, the robustness of the system is an important issue. A good language recognition system is expected to recover language information from various distortions to make a robust decision regardless of speaker and recording environment changes.

1.1 Discriminative cues for language recognition

A fundamental issue of automatic language recognition is to explore the effective discriminative cues to characterize languages. Humans are still the most accurate language identification system in the world today [6]. Within a second of listening to a speech segment, we are able to determine if it is a language that we know. Even if the language is unknown to us, we can often make a subjective judgement about the language, e.g. “It

sounds like Japanese.” In order to have a good automatic language recognition system, we first analyze the features humans use to distinguish languages.

There are a variety of cues that humans use to distinguish among languages [7–9]. In general, they fall into five groups according to their levels of knowledge abstraction: acoustic, prosodic, phonotactic, lexical and syntax. Figure 1.1 illustrates the corresponding features derived from these cues which may be useful for an automatic language recognition system:

Figure 1.1: Five levels of features for language recognition

Acoustic: Acoustic information represents the physical characteristics of the speech signal [1, 10–15], such as the timbre of speech. Typically, the acoustic information of a spoken utterance is represented as a sequence of feature vectors and each individual vector represents the spectral information for a particular time frame. For example, the Mel-Frequency Cepstral Coefficient (MFCC), which is commonly used in automatic speech recognition, is widely adopted as a feature in many language recognition systems.

Prosodic: Prosody refers to the long time acoustic structure [16–19] of an utterance. For example, the stress, intonation (pitch contour), and rhythm (the duration of phones, speech rate) are all elements of the prosodic structure. Compared to acoustic features, which describe frame level characteristics of a signal, the prosodic features capture longer time-span variations across frames.

Phonotactic: Phonotactic information refers to the rules that govern the frequency and combinations of phones in a language [2, 7, 8, 20, 21, 23–25, 27–29]. Phonotactic

rules widely vary from language to language. Although the same phones can be present in many languages, the statistics of these phone occurrences and their combinations are different. It is generally agreed that phonotactics carry more language discriminative information than the phones themselves [30]. Phonotactic information is extracted from the output of a front-end tokenizer which converts a speech segment into a sequence of tokens. Phonotactic features are widely used in language recognition systems because they are easy to obtain yet offer competitive performance. In practice, one or many phone recognizers are used as front-end tokenizer to convert the speech segment into phone sequences.

Lexical and Syntactic: Conceptually, the most important difference between languages is the vocabulary and syntax. With lexical and syntactic information, humans are able to make sense of what they hear in a target language. However, to make use of lexical and syntactic information in language recognition, a language recognition system needs a large vocabulary continuous speech recognition (LVCSR) system for each of the target language as described in [46]. This is impractical in a language recognition task especially when many target languages are involved, as we may not have a LVCSR system for each of the target language.

The differences in the above features must be identified and extracted from raw speech data to solve the automatic language recognition problem. This thesis focuses on the study of phonotactic feature extraction and modeling for language recognition.

1.2 Problem statement and contribution

For a given language, the phonotactic information refers to the constraints of possible sound combinations. We know that the building block of a spoken language is a set of sounds. Although the same sound can be used in many languages, the probability distribution of those sounds and the constraints on the sound combination are different. For example, in English, the sound $/m/$ is commonly observed to appear at the end of words like *swim* and *team*. While in Mandarin Chinese, the sound $/m/$ will never appears at the end of a word. These phonotactic constraints are captured by phonotactic features.

Generally, given an input speech utterance, its phonotactic features can be obtained by following two steps:

- 1) The input speech is first segmented into a sequence of tokens by a front-end tokenizer;
- 2) The n -gram statistics of the tokens are derived, and then used to generate phonotactic features.

The first step to extract phonotactic feature is to establish a front-end tokenizer that can segment a speech utterance into a sequence of tokens. In this thesis, the tokens are referred as sound tokens. We define a ‘sound’ as a phonetically meaningful acoustic segment. It can be a phone or an acoustic segment obtained by a data driven approach [21]. In this thesis, we consider a phone recognizer as a sound tokenizer which uses phones as sound tokens. A phone is usually defined according to phonetic knowledge, and it represents one of the possible sounds in a language.

In this thesis, we propose a method to construct target-oriented phone tokenizers (TOPTs) from a phone recognizer. A phone recognizer is what is defined in speech recognition literature [22]. It consists of a set of phones, and these phone models are trained from data of one or many languages. On the other hand, a target-oriented phone tokenizer is derived from a phone recognizer by taking a subset of phones from it. In other words, the target-oriented phone tokenizer has a phone inventory which is a subset of that of the original phone recognizer. The details of the TOPT will be presented in chapter 4.

This thesis focuses on the study of effective phonotactic feature extraction methods for high performance automatic language recognition. Specifically, the major contributions include:

- A target-oriented method is proposed to construct parallel target-oriented phone tokenizers (TOPT) for robust phonotactic feature extraction.
- A target-aware language models method is proposed as an alternative way to construct front end phone tokenizers.

- A relevance feedback technique is proposed to incorporate more language related information during recognition.
- A feature selection method is explored to eliminate the redundant phonotactic information to make the language recognition system more efficient.

In the next section, I present an overview of the contributions in discriminative phonotactic feature extraction.

1.3 Discriminative phonotactic feature extraction

1.3.1 Target oriented phonotactic feature extraction

It is generally agreed that the accuracy of the phone recognizer greatly affects the performance of the language recognition system [20, 31]. During phonotactic feature extraction, a phone recognizer is used to tokenize the speech utterances of all the target languages. Hence, the phone recognizer needs to make approximations when decoding utterances that are not in the language that it is trained for. In such a language mismatch decoding, the tokenization accuracy will be usually lower than that in matched language cases. Deriving reliable phonotactic features from a low accuracy tokenizer is an important issue in phonotactic based language recognition.

It is also generally agreed that parallel phone recognizers (PPR) can achieve a better language recognition performance than a single phone recognizer (PR) as the decoding front-end [47]. The success of parallel phone recognizers is attributed to the extra informative statistics generated using multiple phone recognizers. Whereas more phone recognizers could be helpful, they also mean that more transcribed data is required to build these phone recognizers. In this thesis, a novel techniques is proposed to enhance the phonotactic features without the need of additional training data, namely target oriented phone tokenizer (TOPT). An TOPT is constructed from a phone recognizer in a target-oriented manner.

TOPT is a novel method to construct a parallel phone recognizers (PPR) front-end for phonotactic feature extraction [33]. For each of the target language concerned, we select a subset of phones from an existing phone recognizer's phone inventory such that

only the phones that have the best discriminative abilities are selected. Each of these phone set will be used to construct a target language oriented phone tokenizer (TOPT). This approach is motivated by the fact that human listeners spot unique sound patterns to identify a language. With the proposed TOPT strategy, more target language specific phone tokenizers can be obtained without requiring additional annotated speech data from new languages.

The idea of TOPT is inspired by the work of bag-of-sounds [26]. The concept of bag-of-sounds is that any spoken language can be represented as the combination of a collection of acoustic segments. A segment can be a phone or any acoustic segment with phonetic meaning. As a sound is a building block of a language, we propose the TOPT method to construct many subsets of the sounds, each of which has a high discriminative ability to distinguish one language from another.

1.3.2 Target-aware language models

In the decoding process, the TOPT is forced to make some approximation on certain sounds that are not covered by the phone set. Such approximation introduces inaccuracy which is adverse to the classifiers. Refining the idea of phone selection, another method of constructing phone tokenizers is proposed: target-oriented language models (TALM).

A target-aware language models (TALM) [34] is an alternative method to derive parallel phone tokenizers from an existing phone recognizer. Different from TOPT method, the TALM method does not make hard decision on the choice of phones in a target-oriented phone tokenizer, but use all the phones with a language model. Each language model is trained to reflect the discriminative ability of individual phones for a specific target language. If the TOPT method is interpreted as making hard decision on the choice of phones, the TALM method can be seen as making soft decision on discriminative phone selection.

1.3.3 Automatic relevance feedback for short utterances

It is observed in recent years' language recognition evaluation (LRE) that the performance of language recognition drops substantially when the test utterance duration is reduced from 30 seconds to 3 seconds [25, 32, 33]. For a phonotactic feature based language

recognition system, the performance degradation may be due to the fact that the phonotactic information obtained from short speech samples is very limited. In light of this, we are prompted to think of ways to increase the amount of language information in test samples by using relevance feedback [35]. The approach operates like this: from development data, we build a reference corpus consisting of all the target languages. The input test speech is then compared with the reference corpus to extract utterances that have similar phonotactic information as the test utterance. These reference utterances are combined with the original test query to form a new test query. In this way, the new test query is expanded with more language information than the original test query.

Three different ways are studied to extract relevant statistics from the reference corpus to augment the initial speech query. The first method expands the original test query with the relevant utterances from the reference corpus. The second method categorizes the reference corpus into clusters, then expands the original test query with the clusters that are the most relevant to the test query. The third approach expands the query with both relevant and irrelevant utterances. The experimental results show that all the three methods improve the performance on the short length language recognition task while they are not effective on longer length language recognition test utterances.

1.3.4 Feature selection to reduce redundant phonotactic information

It has been shown in [25] that significant performance improvement can be achieved by utilizing the phonotactic statistics from the lattice as opposed to using only the single best phone recognition results. The improvement is attributed to the rich information available in the lattice. It has also been shown that the higher order n -gram statistics which contain more discernible phonotactic information will yield a better language recognition system [33, 36]. While these techniques improve the performance of language recognition, their feature dimension is significantly increased and hence requires higher computational cost and a larger amount of training data, which is not always available. The challenge is to reduce the feature dimension while maintaining its discriminative ability.

To solve the problem of handling high dimensional features, a method of reducing the dimensionality of the phonotactic features by eliminating redundant information is proposed. By using this approach, we improve the efficiency of language recognition by using features with smaller dimensions without sacrificing accuracy. With reduced dimensions on lower order n -gram features, we have the capacity of increasing the order of the n -gram statistics to incorporate lexical information to improve language recognition performance. The feature selection is performed by measuring the merit of features according to two different criteria: one is based on the feature's contribution to the language separation margin; the second is the Chi-squared statistics between the feature and language [37]. Using these two criteria, the features are ranked by their merit value, and the features with low merit value are eliminated. Experimental results show that the proposed feature selection method reduces the feature dimension without significant degradation to the language recognition performance.

1.4 Organization of the thesis

The rest of the thesis is organized as follows:

In Chapter 2, the general process of a language recognition system and the performance measurement methods for a language recognition system are introduced. A brief overview of the state-of-the-art language recognition approaches is also presented.

Chapter 3 presents the architecture of a phonotactic feature based language recognition system: PPR-VSM. The experimental setup and corpus used in the experiments are introduced.

Chapter 4 presents the contribution on discriminative phonotactic feature extraction. In this chapter, we present the target-oriented phone recognizer (TOPT) construction method for language recognition.

Chapter 5 introduces an alternative method for target-oriented front end construction: target-aware language models (TALM). This method applies soft decision on the phone selection, hence can be seen as a generalized TOPT method.

Chapter 6 describes a novel method to enhance the input spoken test segments by using the relevance feedback technique to incorporate more language related information on short test segments.

Chapter 7 presents the contribution on phonotactic feature selection. Features that have higher discriminative ability for language identification are selected in the system implementation. The experimental results are reported to evaluate the proposed methods.

Finally, Chapter 8 summarizes the study of the thesis and discusses future directions for research.

Chapter 2

Related Works

Research in automatic language recognition began as early as in 1970s [38]. During nearly 40 years of research, many methods in different aspects were studied to achieve high performance language recognition. In this chapter, we will first introduce the general process required to build an automatic language recognition system. Next we will introduce the performance measure and the NIST language recognition evaluation (LRE) for automatic language recognition systems. And finally, we will present a brief overview on the previous works in automatic language recognition.

2.1 General process of automatic languages recognition

Generally, a language recognition system consists of the following three main components: feature extraction, model training and testing, as illustrated in Figure 2.1.

The feature extraction module converts the continuous speech signal into a sequence of feature vectors. Various features have been studied for automatic language recognition. Although different features capture the language information from different points of view, they share the same objective, that is to extract language related information and eliminate variations introduced subjectively and objectively. Currently, features used in the state-of-the-art language recognition systems can be broadly categorized into the following three categories:

- **Acoustic features:** Phone usage, duration, and language unique sounds are typically reflected in the distributions of spectral characteristics, and are captured in

Figure 2.1: General automatic language recognition process

the acoustic features [1, 10–15];

- **Prosodic features:** stress, pitch contour and rhythm describe long time acoustic structure are captured in the prosodic features [16–19];
- **Phonotactic features:** syllable structure, consonant clusters, and common words represent the lexical constraint in a language, are captured in the phonotactic features [2, 7, 8, 20, 21, 23–25, 27–29].

The process of building models from the training features is known as model training. In the model training process, features derived from the feature extraction module are analyzed to generate one or more models to represent the characteristics of each target language. There are various model training techniques used in language recognition. The simplest form of training uses the feature vectors of the speech segments directly to train the models. Such systems are analogous to pre-defining a set of templates and choosing the template that most closely matches the test pattern. For example, one develops Gaussian Mixture Model (GMM) for spectral feature systems [1]. In phonotactic feature systems, we tokenize the speech into intermediate symbol representation, which is then used to train the classifiers [12]. One example of the phonotactic method is to train a language model for each target language based on phone occurrence statistics. Such a system converts the raw feature vectors into phone sequences and trains the language model according to the occurrence statistics obtained from these phone sequences.

In the testing module, features generated from previously unseen test utterances are evaluated by the trained models. With the evaluation scores, the system makes the language identity decisions [39]. In the testing stage, a suitable decision strategy is applied to meet the application needs. The simplest testing process is straightforward when we deal with a closed-set of languages: the unknown speech feature is evaluated by every target language model and the output scores are compared to determine the language identity. Many approaches do not make decisions on the model output scores directly, but apply score calibration techniques on the output scores to give the final decision.

2.2 Performance measures

This section describes the performance measures for a language recognition system. Generally, there are two types of tasks for an automatic language system: language identification and language verification. Language identification is the process of determining which language is spoken in a given speech segment. Language verification is the process of accepting or rejecting the hypothesis that a given speech segment is from a certain language. The performance of a language recognition system is measured with different measurement criteria according to the task they are performing. The performance of the language identification system is often measured by identification rate (IDR), while equal error rate (EER) and detection cost function (DCF) are used to measure the performance of a language verification task. Nonetheless, a detection error tradeoff (DET) [40] curve is often used to show the overall performance of a language verification system.

2.2.1 Performance measures for language identification

In the language identification task, it is assumed that all the signals are from a fixed set of known languages, thus the language identification task is often referred to as closed-set identification. Assuming there are M target languages, given an unknown speech segment, a language recognition system will give a result that the language identity is m , and $m \in [1, M]$. Until 1999, the language recognition research had mainly focused on the

identification task [9]. The performance of a language identification system is measured with identification rate (IDR) and it is defined as

$$P_{IDR} = \frac{n}{N}, \quad (2.1)$$

where n is the number of correctly identified test utterances and N is the total number of test utterances.

2.2.2 Performance measures for language verification

Recent language recognition research has focused on the language verification task. There are two types of verification tasks: closed-set and open-set verification. In the closed-set language verification task, all the test segments are spoken in languages that are known to the system. In the open-set language recognition task, not all the test utterances' language identities are in the target language set. In other words, no training data from these unseen languages is used to train the language recognition system, and hence no model of these languages exists which could be used for evaluation during testing. Those languages that are not in the target languages are called out-of-set (OOS) languages. By adding a “none-of-the-above” option to the open-set recognition task, one can convert the open-set task to the closed-set language recognition problem.

In a language verification system, given an unknown speech utterance and a target language, the system simply provides a TRUE/FALSE decision to a hypothesis that the speech belongs to a claimed language identity. A common approach of verification system is to set a threshold or an operating point on the score obtained on a given test set for a specific language. If the score for a certain language is above the threshold, the system outputs a “TRUE” decision, otherwise “FALSE”. Two types of errors can be made in the verification process: false alarm (FA) and miss (MS) detection. False alarm or false acceptance occurs when a test utterance is not spoken in the specific target language, but the system gives a TRUE decision. Miss detection or false rejection occurs when a test utterance was spoken in the specific target language but it is not correctly identified by the system.

The performance of the language verification is measured by the combination of these two error rates: false alarm rate (FAR) is the percentage of false acceptance errors for

non-target utterances:

$$P_{FalseAlarm|Non-Target} = \frac{n_{FalseAlarm}}{N_{Non-Target}}, \quad (2.2)$$

Miss rate (MSR) is the percentage of the miss detection errors for target utterances:

$$P_{Miss|Target} = \frac{n_{Miss}}{N_{Target}}, \quad (2.3)$$

For every possible threshold, we will get a pair of FAR and MSR values. The equal error rate (EER) represents the system performance at the operation point when the values of FAR and MSR are equal. The lower the equal error rate value, the better the language verification system.

2.2.3 NIST language recognition evaluation

The NIST language recognition evaluation(LRE) [41] series is organized by the National Institute of Standards and Technology (NIST). The goal of the evaluation is to foster the research in automatic language recognition, and it has the following three main objectives: 1) exploring new ideas in language recognition; 2) developing advanced technology incorporating these ideas; 3) establishing the performance measurement method for language recognition. To date, the NIST Language Recognition Evaluations (LRE) were held in 1996, 2003, 2005, 2007, 2009 and 2011 [41]. The NIST language recognition evaluations have drawn increasing attention from the community through the years. There were only a dozen participating sites in the first evaluation (1996), and since then the numbers have increased to more than twenty in recent years. Recently, NIST LRE is held every other year.

In each year's NIST language recognition evaluation, participants were given a basic data set to train their systems and were also allowed to augment their training data with additional data. However, the extra data must be publicly available and the source of the data must be documented in their system's descriptions. To test language recognition system performance on different lengths of speech data, the NIST LRE test sets are usually separated into 3 categories according to the length of test trials: 30 seconds, 10 seconds and 3 seconds. Actual speech durations vary but they are constrained to

be within the ranges of 2-4 seconds, 7-13 seconds and 25-35 seconds, respectively. The performance is evaluated separately for test segments of different duration.

There have been some changes in each year's evaluation condition. Firstly, more and more languages and dialects have been involved. For example, the 1996 evaluation has 12 target languages and no out-of-set languages, while in 2009, 23 target languages and 5 out-of-set languages are included in the test trials. The first three NIST language recognition evaluations (1996, 2003 and 2005) concentrated on closed set testing. From 2007, more out-of-set languages are included in the evaluation. These unknown languages were not disclosed to participants before evaluation and no training data of them was made available to the participants. The evaluation permitted participant sites to indicate whether their systems were designed primarily for the closed set condition or for open-set condition, or both. Secondly, the testing data condition is also changed in the recent NIST LRE. In the first four NIST language recognition evaluations (1996-2007), the evaluations focus on language and dialect detection in the context of conversational telephone speech, while in NIST LRE 2009 and 2011, in addition to conversational telephone speech, more narrowband conversational broadcast data that came from the Voice of America (VOA) radio broadcasts were included in the evaluation. Please refer to Appendix A for details about the NIST LRE corpus used in this thesis.

The NIST LRE tasks are posed as language verification tasks. Given an unknown speech segment and a language of interest, a language recognition system should make decisions on whether the target language was spoken in the test segment by giving two answers: 1) a hard decision (TRUE or FALSE), and 2) a likelihood score indicating the confidence with which the hard decision is drawn. The higher the score the greater belief that the answer is TRUE.

The performance of an automatic language recognition system will be measured using a detection cost function (DCF) [41], which represents the expected cost of making a detection decision. Suppose the languages under consideration are $l_1, l_2, \dots, l_M, \dots, l_N$, where M is the number of target languages and N is the total number of target languages plus non-target languages. The detection cost function (DCF) for a target language i is

Figure 2.2: DET curve for language recognition

defined as follows:

$$C_{DCF}(i) = (C_{Miss} \times P_{Miss|Target(i)} \times P_{Target(i)}) + \frac{1}{N-1} \sum_{j \neq i} (C_{FalseAlarm} \times P_{FalseAlarm(i)|Non-Target(j)} \times P_{Non-Target(j)}) \quad (2.4)$$

where C_{Miss} and $C_{FalseAlarm}$ represent the relative costs of a miss and a false alarm, respectively, $P_{Miss|Target}$ is the probability of miss for all target tests, while $P_{FalseAlarm|Non-Target}$ is the probability of false alarm for all the non-target tests. P_{Target} is the a priori probability of the target and $P_{Non-Target}$ is $1 - P_{Target}$, which refers to the probability of the non-target. In the NIST LRE, the measurement parameters were set as:

$$C_{Miss} = C_{FalseAlarm} = 1$$

$$P_{Target} = 0.5 \quad (2.5)$$

Smaller value of DCF shows better language recognition performance. Assume the number of detected target languages under consideration is M , we take the overall average

detection cost function as:

$$C_{DCF} = \frac{1}{M} \sum_{i=1}^M C_{DCF}(i) \quad (2.6)$$

A detection error tradeoff (DET) [40] curve is often used to illustrate explicitly the overall performance and the possible error rate tradeoffs between the two error types at each operational point. Figure 2.2 shows an example of a DET curve. The x-axis is the FAR and y-axis is the MSR. Any point on the DET curve representing the DCF can be obtained by using a decision threshold. In Figure 2.2, the circle on the DET curve is the minimal DCF point.

2.3 Review of automatic language recognition techniques

The first study on language recognition started as early as 1970s [38]. In the first twenty years, language recognition research progressed slowly due to the lack of publicly accessible databases. With the increasing demand on multilingual communication and the start of NIST LRE, more significant progress in language recognition has been made in recent years.

In this section, we present a brief overview of the methods used in language recognition according to the features they focused on: phonotactic feature based methods, acoustic feature based methods and prosodic feature based methods. The three categories represent features at different level of abstraction for language recognition. It is found that fusing low level acoustic, prosodic feature and high level phonotactic features greatly improves the performance of a language recognition system [1, 2]. Nevertheless, the most common and successful systems are based on the fusion of multiple features and these systems have shown a good accuracy in the NIST LRE [32, 71, 80, 83].

2.3.1 Phonotactic feature based methods

House and Neuberg [38] made the first attempt of using phonotactic feature for language recognition on text transcriptions of speech data. The text data were converted into sequences of broad phonetic class labels. Those phonetic class label sequences were

modeled by Hidden Markov models (HMMs), and the HMMs were used as classifiers for language recognition. Their experimental results on eight languages showed the feasibility of using phonotactic information for language recognition.

Later in 1980, Li and Edwards [42] extended the idea of broad phone classes to real speech signals. Hidden Markov models were trained from broad phone class labels of six languages. Their experimental results showed that the method can separate Asian languages from Indo-European languages very well.

In 1991, Muthusamy and Cole [43] proposed a language recognition system that utilizes both phonotactic and prosodic features. Two stages of neural networks were used in the modeling process. One set of neural networks was trained from training data and they are used to segment the speech signal into seven broad phonetic classes; another set of neural networks was trained from the phonotactic and prosodic features derived in the first stage and used for language classification.

In 1993, Lamel and Gauvain [44] studied the method of using language specific phone recognizers on language recognition. Unknown speech segments were processed through phone recognizers, the likelihood scores generated from the phone recognizers were compared to make the language decision. This approach was evaluated on a two-language separation problem: English and French. They further extended their work [45] with a two stage method: in the first stage, a single language phone recognizer was used to label multi-lingual speech data. In the second stage, the labels obtained from first stage were used to train several language specific phone recognizers for multiple language recognition. Using the two stages method, the language identification rate improved on the English and French separation.

In 1994, Hazen and Zue [7] developed a language recognition system that uses a single language dependent phone recognizer (PR) as front end to convert speech into phone sequences. An n -gram language model (LM) was trained from phone sequences of each target language, unknown test utterances were evaluated on those language models and the output scores were compared for decision making.

Also in 1994, Zissman and Singer [47] proposed to run parallel phone recognizers (PPR) as the front end to segment speech signals into phone sequences. For every target language, multiple n -gram language models were trained on phone sequences derived

from parallel phone recognizers. A test utterance was evaluated on each of the language models and the scores were compared for the final decision.

In 1997, Navratil et al. [48] described their method of using an HMM based phonetic recognizer to tokenize the spoken signal into phone sequences. The proposed technique had two differences compared to the methods that Hazen and Zissman used. In the decoding process, multiple language specific bigram grammars were used to derive multiple output phone sequences; in language modeling, they applied skip-gram instead of conventional consecutive n -gram language models. The method was evaluated on the NIST LRE 1996 evaluation set. A consistent improvement was achieved on both, the 10 seconds and 45 seconds test sets.

In 2002, Torres et al. [49] proposed to use Gaussian Mixture models(GMM) as front end tokenizers to segment input speech signals into sequences of GMM indices. The GMM index sequences were used for language model training. This method was evaluated on the NIST 1996 evaluation sets. The system based on GMM tokenization achieved competitive results as compared to their phonotactic based system.

In 2004, Gauvain et al. [25] examined a method to extract phonotactic information from the lattice obtained during the phone recognition process. Instead of simply using only the one best hypothesis results for language modeling as proposed in previous methods, the proposed method used phonotactic statistics derived from the phone lattice. The method was evaluated on the NIST 2003 test sets, and obtained significant improvements on the language recognition accuracy. The experimental results showed that richer phonotactic information can be derived from the lattice to achieve a higher language recognition accuracy.

In 2005, Ma and Li [26] proposed a bag-of-sounds method which uses a universal sound recognizer to tokenize an utterance into a sound sequence, and then converts the sound sequence into a count vector, known as bag-of-sounds vector. The method achieves on average 98.1% identification rate (IDR) for 10 seconds test utterances. Later they extended the idea of bag-of-sounds vector to a vector space modeling approach [21], it is one of the state-of-the-art methods in phonotactic feature based language recognition systems [12, 21, 52, 79].

In 2005, Matejka et al. [20] reported the performance of a language recognition system using high quality phone recognizers. Their phone recognizers were trained on long

temporal context features using neural networks. The language recognition system was evaluated on the NIST LRE 2003 data sets, and achieved a very good performance. This suggests that the accuracy of the phone recognizer is crucial to accurate language recognition.

In 2007, Shen et al. [50] proposed to apply speaker adaptation and gender normalization on the phone recognizer to compensate for distortion introduced by the speaker. Speaker adaptation was performed by using the maximum likelihood linear regression (MLLR) method, and gender normalization was achieved by applying vocal tract length normalization (VTLN) in the recognition process. Both methods improved language recognition performance on the NIST LRE 2005 data set.

In 2008, Richardson et al. [36] reported their method of selecting discriminative keywords using recursive feature elimination. The method aims to keep the most discriminative low level n-grams and expand them to higher order n-grams. The effectiveness of this method was evaluated on the NIST LRE 2005 evaluation sets.

In 2009, Marco et al. [51] proposed to use attribute based units instead of phones to tokenize the speech segments for language recognition. In their proposal, spoken language can be described with a universal attribute recognizer which has a common set of fundamental units. The units in the universal attribute set are defined according to the manner and places of articulation of sounds, and are trained with data driven techniques. The recognizer is evaluated on the NIST 2003 test sets and achieved encouraging results.

Recently, Mikel et al. [52] proposed a method that considers cross-decoder phone co-occurrences at the frame level for language recognition. In this approach, phone recognizers were used to perform phone recognition on the input speech, and the frame level phone segmentation is extracted from one best phone decoding. Phone co-occurrences were obtained from the phone segmentations of multiple phone recognizers and were modeled by support vector machine (SVM) classifiers. The proposed method was evaluated on the NIST 2007 test sets, and good performances are obtained.

2.3.2 Acoustic feature based methods

Acoustic features have been used for the language recognition task as early as in 1974 by Leonard and Doddington at Texas Instruments [53]. In the training process, the

training utterances were chunked into segments. For each language, a library of reference segments were derived using a distance measure. During the test process, the sound segments from the testing utterances were compared with the reference segments, and the occurrences of reference matching were calculated to make the decision.

In 1982, Cimarusti et al. [54] conducted a feasibility study on the use of acoustic features for the language recognition task. The linear prediction cepstrum coefficient (LPCC) feature vectors were extracted from each of the input utterances. A polynomial decision function was trained on these feature vectors using an iterative pattern analysis program.

In 1991, Sugiyama [14] proposed to use the vector quantization method to model the LPCC features. In the training process, the LPCC feature vectors of each target language were used to train a codebook of that language. In the recognition process, the input speech was quantized by each codebook and the accumulated quantization distortion was computed. The quantization distortion for all the languages were compared to make the language recognition decision.

Also in the early 90s, Riek et al. [55] proposed to use Gaussian mixture models (GMM) and Nakagawa [74] proposed to use hidden Markov models (HMM) to model the frame based acoustic features. In their systems, the frame based acoustic features of certain target languages were used to train a GMM or HMM model, acoustic features of test utterances were then evaluated on these models to make the language decision.

In the following years, frame based short term spectral information was widely used in language recognition research. A novel feature called the shifted delta cepstral (SDC) feature for language recognition was proposed in 2002 by Torres et al. [11]. The SDC feature is a frame based cepstral feature that captures the spectral information in longer time span. Compared to the short term spectral features like MFCC or LPCC, the language recognition performance was greatly improved by adopting SDC features.

The mismatch between training and testing conditions was one of the focus in language recognition studies. Example of the mismatches are distortions introduced by the changing of speaker, gender and transmission channel. Kenny et al. [56] proposed to use latent factor analysis (LFA) in speaker and language recognition. Alex et al. [57]

adopted the nuisance attribute projection (NAP) method in speaker recognition for language recognition systems [58]. By applying these methods, the language recognition became more robust.

In 2006, Campbell et al. [12] introduced a technique to model the acoustic information using the support vector machine classifier. They proposed a generalized linear discriminate sequence (GLDS) kernel to map the sequential features into a single vector for support vector machines. This kernel solved the problem of comparing acoustic segments that were of different length.

Campbell et al. [59] proposed a Gaussian supervector (GSV) kernel for speaker recognition. In this method, a universal background model (UBM) was trained from acoustic features from all the languages. The training data for each target languages were adapted on the UBM and a language specific Gaussian mixture model was derived for each target language. The means of each Gaussian components of the adapted model were concatenated to form a supervector that represent the discriminative information of the target language. The method was adopted in language recognition and achieved good performance in language recognition.

Burget et al. [60] introduced the discriminative training technique in language recognition. The maximum mutual information (MMI) criterion was used in the model training process on acoustic features. The language recognition performance was greatly improved by adopting this approach.

In 2009, Zhu et al. [61] proposed a discriminative training method for GMM using large margin estimation. Unlike traditional methods, the large margin estimation attempts to enhance the generalization ability of GMM to deal with new data that exhibits mismatch with training data. The method was evaluated on the NIST LRE 2007 data set, and a state-of-the-art language recognition performance was obtained.

2.3.3 Prosodic feature based methods

Currently, the state-of-the-art language recognition systems based on prosodic features obtain a low performance when compared to those systems that use acoustic and phonotactic features. However, prosodic features can be augmented with acoustic and phonotactic features [2] to improve the language recognition performance.

In 1986, Foil et al. [62] explored the usage of prosodic features in language identification. He conducted his experiments on noisy data recorded from the radio using pitch and energy contours. In their experiments, three languages, each from a different major language group, were used for development and testing. An identification rate of 64% is obtained with short speech segments, which showed the feasibility of using prosodic feature in language recognition.

In 1991, Savic et al. [63] studied a language identification system utilizing language specific pitch contours. Experimental results showed that the pitch contours convey discriminative information for language recognition.

In 1993, Yeshwant [64] used speech rate and syllable timing information in his language identification system. However, according to his experimental results, the prosodic features provided limited improvements in his system.

In 1996, Thyme-Gobbel et al. [65] conducted pair-wise language identification on English, Spanish, Japanese and Mandarin using a set of prosodic features like: pitch, pitch contour, amplitude, rhythm etc. Their results showed that the pitch based features are good in distinguishing some language pairs while amplitude related features were less effective in language separation.

In 1999, Ramus et al. [66] explored the discrimination ability of languages using the re-synthesis technique. The experimental results revealed that syllabic rhythm and intonation were important cues for language separation and the syllabic rhythm is the most important cue for French speakers to discriminate English from Japanese sentences.

In 1999, Cummins et al. [67] applied recurrent neural networks to model two types of prosodic features: delta-F0 and the amplitude envelope. Their experimental results were consistent with Thyme-Gobbel's in that the effectiveness of prosodic feature vary on different target languages.

In 2003, Adami et al. [16] proposed to use the temporal trajectories of the fundamental frequency and short-term energy to characterize speaker and language. The speech signal was segmented into discrete prosodic class units, n -gram modeling of segment classes were used for language decision. The experiments showed that the prosody dynamics may characterize some languages better than the phonotactic based approach, for example, Chinese.

In 2005, Obuchi et al. [17] proposed to use prosodic HMMs in language recognition. The power, F0 and the reliability of the F0 are used. A data driven clustering method was used to create language dependent prosodic HMMs. The prosodic system has two advantages compared to the phonotactic system: it requires no labels to create HMMs and it is computationally cheaper.

In 2006, Ma et al. [68] proposed a method to extract tone relevant features based on pitch flux. The auto correlations of two adjacent frames were calculated and the covariance between them was estimated to extract pitch flux features. The tone features, combined with MFCC features were evaluated in a three Chinese dialect recognition task. The experimental results showed that the pitch flux features were very effective on short test utterances of tonal language.

In 2008, Leena et al. [18] proposed a new approach to extract and represent prosodic features. The vowel onset points were used to segment the speech signal, while the F0 and energy contour were estimated in each consecutive segments. The algorithm was evaluated on the NIST LRE 2003 test sets and showed the potential of prosodic features to distinguish languages.

In 2009, Yin et al. [19] proposed to segment speech signals using articulatory features like voicing status. A pair of connected unvoiced and voiced segments was considered as the sound unit, the statistics of these sound units were learned by n -gram models for language recognition. By fusing the proposed duration modeling based system with GMM-UBM based acoustic system, a relative 19.7% EER reduction on the NIST LRE 2005 language recognition was obtained. The experiment confirms that prosodic information is complementary to the acoustic feature in language recognition.

2.4 Summary

In this chapter, we have discussed the main components of an automatic language recognition system and its performance measure methods. We also introduced the NIST language recognition evaluation framework in which we will evaluate our proposed methods.

Furthermore, we have also provided a brief overview of previous work in automatic language recognition. They are grouped into three categories according to features they

are focused on: acoustic feature based system, phonotactic feature based system and prosodic feature based system. In practice, a typical language recognition system is based only on one type of feature, while in recent LREs, it is more common to use a fused system which leverages over different complementary features. A fused system takes advantages of discriminative information provided from different features, and therefore usually offers better language recognition performance than a single feature system.

In the next chapter, we will concentrate on phonotactic feature based language recognition methods and our baseline system architecture PPR-VSM.

Chapter 3

Phonotactic Feature Based Language Recognition

The focus of this thesis is on phonotactic characterization to capture the lexical constraints and phonotactic patterns for automatic language recognition. The first step towards phonotactic characterization is to establish a front-end tokenizer, to convert the input speech into a sequence of tokens. In this thesis, the tokens are referred as sound tokens. We define a 'sound' as a phonetically meaningful acoustic segment. It can be a phone or an acoustic segment obtained by a data driven approach [21]. In this thesis, we consider a phone recognizer as a sound tokenizer which uses phones as sound tokens. A phone is usually defined according to phonetic knowledge, and it represents one of possible sound in a language.

The following two phonotactic feature extraction and modeling approaches are commonly used in the state-of-the-art automatic language recognition systems.

The first approach is to use parallel phone recognizers followed by language modeling (PPR-LM). In this framework, the phonotactic information is represented by phone n -gram language model. For each target language, a language model (LM) is trained from phone n -gram statistics of that language. The test segment's phone sequence is estimated by all candidate languages' language model and the language with the highest likelihood is selected as the language identity of the test speech segment [7, 8, 20, 47].

The second approach is to use a parallel phone recognizer followed by vector space modeling (PPR-VSM). In this framework, the phonotactic features are extracted by

converting the phone n -gram statistics into high dimensional feature vectors. SVM classifiers are used to model the difference between languages [26, 85, 86]. A test segment is evaluated on the SVM classifiers to make its language identity decision.

In this chapter, we first present an introduction of two techniques which use phone recognizer followed by language modeling for language recognition: PR-LM and PPR-LM. We then introduce the architecture of our baseline system: PPR-VSM language recognition system.

3.1 Techniques for phonotactic feature based language recognition

As introduced in the previous chapter, Hazen et. al [7] were the first to use a phone recognizer followed by language modeling (PR-LM) for language recognition. Later Zissman et.al [47] extended the PR-LM to PPR-LM by incorporating more phone recognizers as the tokenization front-end. In this section, we describe the PR-LM and PPR-LM approaches in detail.

3.1.1 Phone recognizer as front-end

3.1.1.1 PR-LM

A PR-LM system consists of a language independent phone recognizer followed by a language dependent language modeling module [7]. Figure 3.1 illustrates the structure of a standard PR-LM language recognition system.

Assuming there are M target languages: $l_1, l_2, \dots, l_M$, a PR-LM system consists of a single phone recognizer and M language models. During phone recognition, an utterance is decoded by the phone recognizer into a phone sequence: $S = \{t_1, t_2, t_3, \dots, t_i, \dots\}$, where t_i denotes the i^{th} phone in the sequence. A language model λ_m is trained for target language l_m using the phone sequences for that language. For example, to train an English language model, a collection of speech data in English is processed by the phone recognizer to generate a set of phone sequences. This set of phone sequences is used to train the English language model (LM) [7, 8]. For a language recognition task of M target languages, M such language models are trained.

Figure 3.1: PR-LM system with M target languages

The language model is an important module in a statically-based speech recognition system. A language model is trained from an existing text corpus of a specific language, and it is used by the speech recognizer to estimate how likely a word sequences is, independent of the acoustic information. N -gram language models are commonly used in automatic speech recognition and have been adopted in automatic language recognition applications. Generally, an n -gram is a subsequence of n items from a given sequence. This sequence can be phones, syllables, words etc. Specifically, an n -gram of size 1 is referred to as a unigram; size 2 is a bigram; size 3 is a trigram; and others are simply called n -gram. For example, given a speech utterance, if the output of a phone recognizer is: “sil th ee s sil ie z sil aa sil t eh s t sil”, the following n -gram statistics can be obtained from this phone sequence:

- 1-gram (unigram): sil(5), th(1), ee(1), s(2), ie(1), z(1), aa(1), eh(1), t(1)
- 2-gram (bigram): sil th(1), th ee(1), ee s(1), s sil(1), sil ie(1), ie z(1), z sil(1),...
- 3-gram(trigram): sil th ee(1), th ee ss(1), ee ss sil(1), ss sil ie(1), sil ie z(1), ...

A n -gram model is trained from the n -gram statistics of training corpus. The n -gram model can then be used as a probabilistic model to predict the next item in a sequence. In the training process of the n -gram model, the statistical independence assumption of

the phone elements is made. For a unigram model, the probability of each phone element is assumed to be statistically independent of all other phone elements. That is:

$$P_u(S|l_m) = P(t_0, t_1, \dots, t_{k-1}, |l_m) = \prod_{j=0}^{k-1} P(t_j|l_m) \quad (3.1)$$

where $S = t_0, t_1, \dots, t_{k-1}$ is a phone sequence with k elements, t_j denotes the j^{th} phone element, $P(t_j|l_m)$ and $P(S|l_m)$ are the occurrence probabilities of the phone element t_j and phone sequence S in target language l_m , respectively.

In a bigram model training process, we assume that each phone is dependent only on its preceding phone, i.e.,

$$P_b(S|l_m) = P(t_0|l_m) \prod_{j=1}^{k-1} P(t_j, t_{j-1}|l_m) \quad (3.2)$$

Similarly, in a trigram model, we assume that each phone is statistically dependent only on the two preceding phones, i.e.,

$$P_t(S|l_m) = P(t_0|l_m)P(t_1|l_m) \prod_{j=2}^{k-1} P(t_j, t_{j-1}, t_{j-2}|l_m) \quad (3.3)$$

In practice, different order of LMs are usually interpolated to take advantage of different degree of n -grams. For example, an interpolated LM that consists of unigram, bigram and trigram models:

$$P(S|\lambda_m) = \alpha_0 + \alpha_1 P_u(S|l_m) + \alpha_2 P_b(S|l_m) + \alpha_3 P_t(S|l_m) \quad (3.4)$$

where α_i ($i \in [0, 3]$) are obtained by the criterion of minimizing the perplexity [8] of the training corpus.

In a PR-LM language recognition system, the training data from the target languages are converted into phone sequences by the phone recognizer. Those phone sequences from same target language l_m are used to train an interpolated language model λ_m as defined in Equation 3.4. For M target languages, M such language models are trained. During testing, an unknown speech segment is first passed through the phone recognizer to generate a phone sequence S . The likelihood scores of S on these M language models are estimated and compared. The unknown utterance is then classified as the language that gives the highest likelihood score.

$$l^* = \operatorname{argmax}_{m \in [1, M]} P(S|\lambda_m) \quad (3.5)$$

3.1.1.2 PPR-LM

 Figure 3.2: PPR-LM system with F phone recognizers and M target languages

Another type of language recognition system is called the Parallel Phone Recognition followed by Language Modeling (PPR-LM) system. The PPR-LM method was extended from PR-LM method by using many phone recognizer [47] as opposed to a single recognizer. It is believed that different phone recognizer may present different aspect of phonotactics. Figure 3.2 illustrates the architecture of a PPR-LM system.

Compared with the PR-LM system, the PPR-LM system consists of F ($F > 1$) phone recognizers instead of just one. Suppose we have F phone recognizers as the tokenization front-end. During phone recognition, an utterance is decoded by these phone recognizers into F independent sequences of phone tokens $S = \{S_1, S_2, \dots, S_F\}$, where S_f denotes the phone sequence generated from the f^{th} recognizer,

$$S_f = \{t_{f,1}, t_{f,2}, t_{f,3}, \dots\} \quad (3.6)$$

where $t_{f,*}$ is a phone of the f^{th} phone recognizer. In the training process, for each phone recognizer, M language models are trained following the same process as described in the PR-LM system. Hence the PPR-LM system has $F \times M$ language models for M target languages: $\{\lambda_{11}, \lambda_{12}, \dots, \lambda_{1M}, \lambda_{21}, \dots, \lambda_{2M}, \dots, \lambda_{F1}, \dots, \lambda_{FM}\}$.

During the testing phase, an unknown speech segment is first processed by the F phone recognizer to generate F phone sequences. The likelihood of each phone sequence is evaluated by the M language models trained using the corresponding language recognizer. Hence $F \times M$ language scores are generated as illustrated in Figure 3.2 where $p(S_f|\lambda_{fj})$ denotes the language model likelihood score obtained by testing phone sequence S_f on language model λ_{fj} . These scores are often normalized to compensate for the score bias caused by using different phone recognizers [1]. Many pattern classification methods can also be applied to the $F \times M$ corresponding language scores to make the final language recognition decision [2].

Although the computational effort of the PPR-LM system is significantly higher than PR-LM system, much better accuracy was achieved [8]. Currently, the PPR-LM system is one of the most established method for language recognition.

3.1.1.3 Phonotactic based language recognition using lattice

In a phonotactic based language recognition system, one or more phone recognizers are used to convert speech segments into phone sequences. The common implementation is to derive phonotactic features from the best phone sequences obtained from phone recognition. As many approximations need to be made to generate the best phone sequence hypothesis, the accuracy of the phone sequences may be degraded. To alleviate this problem, Gauvain et. al [25] proposed to derive phonotactic information from phone lattice instead of simply using the best hypothesis.

A phone lattice is a graph that shows the possible phone paths among the feature frames, an example of phone lattice of word ‘seven’ is illustrated in Figure 3.3. For each edge, the hypothesized phone and its acoustic likelihood score are presented. In the phone recognition process, it is very likely that the same phone is included in more than one path, the probabilities of a given phone at a given position over all possible paths are captured in a lattice. The phonotactic information extracted from the lattice is obtained by summing the likelihood over all paths in the phone lattice.

Figure 3.3: Phone lattice of word ‘seven’ generated by an English phone recognizer

Compared with the best hypothesis, a lattice contains richer language information. By using the lattice to extract the phonotactic information, the performance of the language recognition system is improved. The lattice based phonotactic feature extraction showed significant improvements in the language recognition performance and it is currently one of the most successful algorithms used for language recognition [25, 32, 33].

3.2 PPR-VSM language recognition system

In this section, we introduce a parallel phone recognizer followed by the vector space modeling (PPR-VSM) method for automatic language recognition [21]. The method is similar to the PPR-LM method in the front-end, as it uses parallel phone recognizers to tokenize the speech segments into multiple phone sequences. The difference is the PPR-VSM method adopts vector space modeling instead of language modeling in the back-end stage.

The vector space modeling (VSM) method has been widely used in information retrieval since its introduction around thirty years ago [69]. In the vector space modeling approach for information retrieval, both documents and queries are represented by vectors containing the statistics of the term occurrences. By representing a document with a term occurrence statistics vector, a continuous degree of similarity between vectors can

be easily obtained for classification or ranking. Discriminative training techniques can also be applied to those vectors to achieve a higher retrieval accuracy. Inspired by the success of vector space modeling for information retrieval, we apply it to the automatic language recognition task [21, 73].

Figure 3.4: PPR-VSM language recognition system with F phone recognizers

Figure 3.4 illustrates the architecture of a PPR-VSM language recognition system. There are three components in the PPR-VSM system [21]: PPR front-end, VSM back-end and recognition. In a PPR-VSM language recognition system, a collection of parallel phone recognizers (PPR) serve as the tokenization front-end to generate phone sequences for the vector space modeling (VSM) back-end. In the recognition component, the language classification is carried out based on the composite n -gram occurrence vector. We will describe the details of each component in the following sections.

3.2.1 VSM back-end

For each generated phone sequence S_f (Equation 3.6), it can be expressed as an n -gram statistic vector:

$$\mathbf{v}_f = [\mathbf{v}_{f,1} \ \mathbf{v}_{f,2} \ \dots \ \mathbf{v}_{f,n}] \quad (3.7)$$

where $\mathbf{v}_{f,n}$ stands for the n -gram statistic vector with n -gram of order n . Specifically, $\mathbf{v}_{f,1}$ is the unigram statistic vector,

$$\mathbf{v}_{f,1} = [p_{f,1} \ p_{f,2} \ \dots p_{f,3} \ \dots p_{f,e_f}] \quad (3.8)$$

each element $p_{f,i}$ denotes the probability of the i -th phone of the f -th phone recognizer, $t_{f,i}$, occurs in the phone sequence S_f , e_f denotes the number of phones in the f^{th} phone recognizer.

$$p_{f,i} = \frac{\text{count}(t_{f,i}|S_f)}{\sum_{j=1}^{e_f} \text{count}(t_{f,j}|S_f)} \quad (3.9)$$

Similarly, $\mathbf{v}_{f,2}$ is the bigram occurrence vector:

$$\mathbf{v}_{f,2} = [p_{f(1,2)} \ p_{f(2,3)} \ \dots p_{f(j-1,j)} \ \dots] \quad (3.10)$$

where element $p_{f(i-1,i)}$ stands for the joint probability of the bi-phone $t_{f,i-1}t_{f,i}$ in the phone sequence S_f :

$$p_{f(i-1,i)} = \frac{\text{count}(t_{f,i-1}, t_{f,i}|S_f)}{\sum_{j=1}^{e_f} \text{count}(t_{f,j-1}, t_{f,j}|S_f)} \quad (3.11)$$

In this way, the phone sequence generated from f^{th} phone recognizer S_f is converted into an n -gram occurrence vector $\mathbf{v}_f$ (Equation 3.7). Furthermore, we can concatenate the occurrence vectors from the F phone recognizers into a high dimensional composite feature vector:

$$\mathbf{v} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_f \ \dots \ \mathbf{v}_F] \quad (3.12)$$

The dimension of the feature vector is then equal to the number of n -gram patterns. For example, if the unigram and bigram are considered, the dimension of the phonotactic composite vector $\mathbf{v}$ will be:

$$d = \sum_{f=1}^F e_f + e_f^2. \quad (3.13)$$

3.2.2 Recognition

The recognition module uses SVM classifiers to reduce the dimensionality of the input n -gram statistic vectors to lower dimensional discriminative feature vectors. Specifically, for each target language, an SVM classifier is trained using the composite feature vectors

of the target language as the positive set, and the composite feature vectors of all other languages as the negative set. With M target languages, we will build M such SVMs. The output scores of these M SVMs are used to produce an M -dimensional discriminative vector [2]. In this way, we have projected a very high dimensional feature vector into a much lower dimensional one. These M -dimensional discriminative feature vectors are then used as the input feature vectors for language recognition.

We formulate the language recognition as a hypothesis test. For each target language, we build a language detector consisting of two Gaussian mixture models (GMMs) l^+ and l^- . l^+ is trained on the M -dimensional discriminative vectors of target language, called positive model, while l^- is trained on those vectors of its competing languages, called negative model. We define the confidence of a test sample O belonging to language l^+ as the posterior odds in a hypothesis test under the Bayesian interpretation. We have H_0 which hypothesizes that O is language l^+ , and H_1 , which hypothesizes otherwise. The posterior odd is approximated by the likelihood ratio λ that is used for the final language recognition decision,

$$\lambda = \log \frac{P(O|l^+)}{P(O|l^-)} = \log P(O|l^+) - \log P(O|l^-) . \quad (3.14)$$

Equation 3.14 gives a relative log-likelihood score between the target language and its competing languages. In the language recognition process, for a target language claim, if the λ value of the test utterance is higher than a predefined threshold, the system returns a TRUE answer, otherwise FALSE.

3.2.3 PPR-LM versus PPR-VSM

The PPR-LM and PPR-VSM approaches are similar in the tokenization front-end which converts speech segments into phone sequences. Their difference lies in the language classifier training methods. The PPR-LM approach uses a generative way to train the language classifiers, i.e., the n -gram statistics are used to capture the desired language characteristics, while in the PPR-VSM method, discriminative language classifiers are trained by using not only the n -gram statistics of the desired target language but also the n -gram statistics of the rest competing target languages.

According to the experimental results reported in [21], the VSM back-end demonstrates a clear advantage over the LM back-end for long test utterances, while LM works better for short test utterances, for example, the 3 seconds test utterances in the NIST LRE. This may be due to the fact that the VSM model is designed to capture higher order phonotactics through the SVM training process as it takes the discriminative characteristic among languages into consideration. While in the LM training process, only the target language characteristic is considered.

3.3 Summary

In this chapter, we have focused our discussion on phonotactic feature based language recognition approaches. Two phonotactic feature based systems are introduced: PPR-LM and PPR-VSM. The methods use the same tokenization front-end but different ways of representing and modeling the phonotactic feature. The two methods achieve comparable performances over the given tasks.

The phonotactic information can also be derived from the lattice, which is common practice in many state-of-the-art language recognition systems. Compared with the best phone recognition hypothesis, a lattice contains richer language information. The lattice based phonotactic feature extraction showed significant improvements, thus becoming one of the most successful implementations in phonotactic based language recognition.

In the next chapter, we present our proposed discriminative phonotactic feature extraction approach that furthers the abovementioned techniques.

Chapter 4

Target-oriented Front End for Automatic Language Recognition

4.1 Objective

One challenge of the language recognition task is to identify the discriminative cues of spoken languages and to effectively organize these language cues in the classifier design. The most widely adopted method to extract phonotactic cues is to use parallel phone recognizers as the front-end. Experimental results show that language recognition performance improved with the use of higher order n -gram statistics [21] or when a larger number of phone recognizers is used [8]. Intuitively, higher order n -grams contain longer phonetic statistics which may convey lexical rules of the language, and hence they would be more discriminative than lower order n -grams. In the case when a larger number of phone recognizers is used, they will provide more perspectives of the phonotactic statistics which are beneficial for language recognition. However, in practice, the number of phone recognizers and the order of n -gram statistics are bounded by the availability of training samples and the capability of modeling tools. We are motivated to find a solution to increase the number of phone recognizers and the order of n -gram statistics without the need to significantly increase the number of training samples.

In this chapter, we present our approach to improve the discriminative performance of the parallel phone recognizers (PPR) front-end using a target-oriented method. Our approach constructs a set of new phone tokenizers from existing phone recognizers without requiring new transcribed training data. The methods presented in this chapter have

contributed to the Institute for Infocomm Research (IIR)’s submission to the NIST LRE 2007 and 2009.

4.2 Target-oriented phone tokenizer

A phone recognizer in the phone tokenization front-end is not necessarily trained from one of the target languages as long as it can generate informative phone sequences. For example, an English phone recognizer may be seen as a listener with English background. When the listener identifies a segment of speech in unknown language, the speech input will be expressed as an English phone sequence with phonotactic statistics that are different from English.

We believe that the discriminative ability of the PPR front-end can be improved by optimizing it to the set of target languages by reconfiguring them as target-oriented phone tokenizers (TOPT). For example, given an English phone recognizer, we can create an Arabic-oriented English phone tokenizer or a Mandarin-oriented English phone tokenizer. Our motivation stems from the observation that for different languages, e.g., Arabic or Mandarin, their phonotactic statistics would be different to an English listener. Although all the phones of the English recognizer may be useful, not all the phones and their phonotactics are equally significant. We therefore propose a target-oriented front-end tokenizer construction method which assumes that the phones of each target-oriented phone tokenizers can simply be a subset of the original recognizer’s phone inventory that can best discriminate the target languages.

To differentiate a target-oriented phone tokenizer from a conventional phone recognizer, we make a distinction between phone recognizer and phone tokenizer. A phone recognizer consists of a set of phones that are defined by phonologists, and the phone models are trained from data of one or many languages; while a phone tokenizer is derived from a phone recognizer by taking a subset of phones from the original phone recognizer. In other words, the phone tokenizer has a phone inventory which is a subset of that of the original phone recognizer.

The proposed target-oriented front-end optimization strategy has the following three desirable properties:

- A target oriented phone tokenizer (TOPT) capture the characteristics of the target language by using a unique subset of phones of a given phone inventory;
- We can increase the number of phone tokenizers without requiring additional training samples as many different phone subsets can be drawn from a single phone recognizer;
- The use of higher order n -gram phonotactic statistics become feasible as TOPT usually has a relatively smaller phone inventory than the original phone recognizer.

Figure 4.1 illustrates the proposed target-oriented phone tokenizers construction method. Given an phone recognizer, we propose the following two-step strategy to derive multiple target language oriented phone tokenizers:

Figure 4.1: Target-oriented phone tokenizer

In the first step, we select a subset of phones to construct a new phone tokenizer for each target language. Assume we have a phone recognizer with phone inventory of n phones. We first estimate the discriminative ability of each phone t_i for each target language. By choosing discriminative phones for each target language, we construct an optimized phone tokenizer for every target language. With a subset of phones from the

original phone recognizer, all sounds from the input speech will be forcibly decoded to this reduced phone set in the decoding process. For M target languages, we may construct M candidate target-oriented phone tokenizers $\{TOPT_1, TOPT_2, \dots, TOPT_M\}$, one for each target language.

In the second step, we will shortlist some of the M TOPT candidates as the desired PPR front-end for language recognition. As all the TOPT candidates are generated from the same phone recognizer, it is possible that TOPT candidates have highly overlapped phone sets. Hence, redundant information may be obtained if we use similar TOPT candidates in the PPR front-end. One way to reduce the redundancy is to keep only representative TOPTs. The challenge is to quantitatively measure the usefulness of each of the TOPT candidates. We hence propose a method to select the TOPT candidates by measuring the merit of each TOPT candidate. The merit value takes account of the discriminative ability and distinctiveness of an TOPT. The details are given in the following sections.

4.2.1 Phone selection strategy

To determine the importance of a phone for recognizing a language, we propose two criteria to measure the discriminative index of a phone: 1) the phone's contribution to the separation margin (SM) in a support vector machine (SVM) classifier, and 2) the phone's mutual information (MI) for target languages.

4.2.1.1 Separation margin (SM)

In this criterion, we measure each phone's contribution to the language separation margin of an SVM classifier. We adopt the unigram statistics of phones as the feature vector to construct a one-versus-rest linear SVM hyperplane to separate a target language from the rest of languages (non-target languages).

Assume we have a phone recognizer with e phones: $\{t_1, t_2, \dots, t_e\}$. To identify M target languages: $l_1, l_2, \dots, l_M$, we first convert training utterances into phone sequences using this phone recognizer. Each phone sequence S can be converted to an e -dimensional feature vector $\mathbf{v}$ using only the unigram statistics of each phone in the same way as in Equation 3.8.

$$\mathbf{v} = [p_1 \ p_2 \ \dots p_3 \ \dots p_e] \quad (4.1)$$

For each target language l_m , an SVM is trained using the unigram feature vectors derived from the target language as the positive set and the feature vectors from all the rest of languages as the negative set.

An SVM [92] classifier learns a binary decision over a set of feature vectors $\mathbf{v}$ in the form of

$$f(\mathbf{v}) = \alpha^T \phi(\mathbf{v}) + b, \quad (4.2)$$

here α is a weight vector, b is the offset vector, and $\phi(\cdot)$ is a kernel function. Given a feature vector $\mathbf{v}_j$, when $f(\mathbf{v}_j) > 0$, we classify the speech segment as the target language, otherwise, we classify it as non-target language.

The SVM learning is posed as an optimization problem with the goal of maximizing the separation margin, i.e. the distance between the separating hyperplane $\alpha^T \phi(\mathbf{v}) + b = 0$ and the nearest training vectors. Let the i -th element of α be denoted as α_i . The procedure to select a phone is to estimate the importance of each feature element by examining how it influences the width of the margin in the resulting hyperplane. It was found that the margin is inversely proportional to $\|\alpha\|$, i.e. the length of α [92]. Hence if a feature element p_i has higher weight α_i , it is more influential in determining the width of the separation margin. In other words, the corresponding phone t_i is more expressive with respect to the target language.

4.2.1.2 Mutual information (MI)

In a different approach for phone selection, we measure the importance of a phone using information theory. We consider the identity of a language l as a random variable that can take two values in $C = \{l^+, l^-\}$, where l^+ and l^- denote target and non-target language respectively. We use i to denote one of the phones in the existing phone recognizer's phone inventory. For a given language l , the presence of phone i is another random variable that takes two possible values in $T = \{t_{i+}, t_{i-}\}$, where t_{i+} denotes that the phone i is present in target language and t_{i-} denotes the phone i is not present in the target language.

The mutual information of the phone presence T and language category C can then be estimated by

$$MI(T, C) = \sum_{t_i \in T} \sum_{l \in C} p(t_i, l) \log \frac{p(t_i, l)}{p(t_i)p(l)} \quad (4.3)$$

where $p(t_i, l)$ is the prior probability of phone i appears in language l , $p(t_i)$ and $p(l)$ are the probability of t_i and language l respectively. Those phones with higher mutual information value are more expressive than others for a given target language.

4.2.2 Tokenizer selection

Based on the phone selection criteria as described in Equation 4.2 and Equation 4.3, a subset of phones that have higher discriminative ability can be selected and used to construct a new phone tokenizer. Assuming there are M target languages, we may construct M target-oriented phone tokenizer (TOPT) candidates for each phone recognizer. Given F phone recognizers, we can generate $F \times M$ TOPT candidates. While more parallel tokenizers may not harm the system, we may wish to reduce the number of TOPT candidates in the front-end due to two reasons: first, the phones in the TOPT candidates are highly overlapped as they are derived from the same phone recognizer. Hence if we use all the TOPTs as tokenization front-end, there may be a lot of redundant information; second, more parallel tokenizers mean a higher dimension of feature vectors, and hence a higher computational cost and more data are required in training and testing. In view of this, we should select TOPT candidates based on their discriminative ability and distinctiveness.

In this section, we discuss how to evaluate the discriminative ability and distinctiveness of a tokenizer.

4.2.2.1 Discriminative ability of tokenizers

In the PPR-VSM framework, the speech data from different languages are first converted into a set of phone sequences by multiple phone tokenizers. The phone n -gram statistics obtained from these phone sequences are used to model the languages. To evaluate the discriminative abilities of the TOPT candidates, one way is to evaluate how much language related information is captured in the n -grams derived from the TOPT candidates.

In information theory, the conditional entropy quantifies the uncertainty of a random variable given that another random variable is known. It is adopted in language recognition research to measure the relationship between features and languages [77, 93]. We propose to estimate the discriminative ability of the TOPT candidates using the conditional entropy of n -gram statistics given a set of target languages.

Let L denote a set of target languages $L = \{l_1, l_2, \dots, l_M\}$, where M is the number of target languages. Suppose $\pi_k = \{T_1, T_2, \dots, T_{d_k}\}$ denotes a set of n -gram terms that can be derived from the k^{th} target-oriented phone tokenizer $TOPT_k$, and d_k is the number of possible n -grams for the k^{th} TOPT according to Equation 3.13. The conditional entropy of the set of n -grams to the language set L can be calculated by

$$H(\pi_k|L) = - \sum_{m=1}^M \sum_{i=1}^{d_k} p(T_i|l_m) \log p(T_i|l_m) \quad (4.4)$$

The conditional entropy given in Equation 4.4 measures the uncertainty of the distribution of the n -gram statistics among different languages. Lower conditional entropy can be interpreted as the evaluated set of n -grams having less uncertainty for deciding the language identity. The TOPT candidate with lower conditional entropy have higher discriminative ability for language recognition. In this way, the TOPT candidates can be ranked according to their discriminative abilities: $H(\pi_k|L)$.

4.2.2.2 Distinctiveness of tokenizers

In this subsection, we propose a measure to determine the distinctiveness of an TOPT tokenizer. Given M target languages, a set of M TOPT candidates can be derived from a phone recognizer which has N phones in the phone inventory. Each TOPT candidate contains a subset of phones in the phone inventory. The phones in each TOPT can be represented by a N dimensional binary code vector:

$$[o_{k,1} \ o_{k,2} \ \dots \ o_{k,i} \ \dots \ o_{k,N}]^t \quad (4.5)$$

where $o_{k,i}$ is 1 if t_i is present in the k^{th} TOPT candidate and 0 otherwise. With M such binary code vectors derived from M TOPT candidates, we can calculate the pair-wise

Hamming distances among these binary code vectors. The averaged pair-wise Hamming distance represents the distinctiveness of the TOPT:

$$AH_k = \frac{1}{M-1} \sum_{j \neq k}^M \sum_{i=1}^N |o_{k,i} - o_{j,i}| \quad (4.6)$$

where $|o_{k,i} - o_{j,i}|$ is 0 if $o_{k,i}$ and $o_{j,i}$ have the same value, and 1 otherwise. The value of AH_k measures the distinctiveness of the k^{th} TOPT candidate. In this way, a high AH_k value suggests that the k -th TOPT is highly distinctive.

4.2.2.3 Merit of tokenizers

We combine the discriminative (Equation 4.4) and distinctiveness measure (Equation 4.6) to introduce the following merit for TOPT candidates selection.

$$D_k = \frac{AH_k}{H(\pi_k|L)} \quad (4.7)$$

A suitable target-oriented tokenizer should have a high distinctiveness value AH_k and its n -gram statistics should have low uncertainty on the target languages $H(\pi_k|L)$. In this way, we rank the TOPT candidates derived from the same phone recognizer. An TOPT candidate with high merit value means it has high discriminative ability in separating languages, and at the same time it is not similar as other TOPT candidates.

As the distinctiveness value changes with the changing of the set of the TOPT candidates that it is evaluated on, we propose an iterative approach in TOPT selection. We denote the original set TOPT candidates as ψ_s , it contains all the TOPT candidates before selection. The set of target TOPTs that are chosen as PPR front-end is denoted as ψ_e , it is empty at the beginning of the selection. The iterative TOPT selection process is performed in the following steps:

- 1). For each TOPT in ψ_s , calculate their merit value as defined in Equation 4.7. The TOPT which has the highest merit value is selected and moved to ψ_e .
- 2). For each TOPT in ψ_s , recalculate their distinctiveness value according to TOPT in ψ_e , i.e., the average Hamming distances with TOPTs in ψ_e can be obtained using Equation 4.5. According to the new distinctiveness value, we update the merit value of TOPTs in ψ_s , the TOPT with the highest merit value is moved to ψ_e .

3). Repeat the step 2) until we have desired number of TOPTs.

In this way, we can obtain a set of TOPTs which have high discriminative ability and they are distinctive to each other. In the next section, we report our experimental results on the proposed TOPT selection method.

4.3 Phone Recognizers

We will validate the proposed target-oriented phone tokenizer approach with a series of language recognition experiments. Before going to the details of experiments, we introduce the phone recognizers used in the experiments. Two sets of language specific phone recognizers and three universal phone recognizers are used as front-end recognizers to convert the speech segments into phone sequences.

Table 4.1 lists some the language specific phone recognizers and universal phone recognizers used in the experiments. The table also indicates their training corpus, the number of phones and Gaussian components used in each hidden Markov model (HMM) state. The amount of training data used for seven language specific phone recognizers are also reported.

4.3.1 Language specific phone recognizers

The first set of phone recognizers, PPR1, consists of seven language specific phone recognizers. The languages are English, Korean, Mandarin, Japanese, Hindi, Spanish and German. A total of 300 phones, 44 for English, 37 for Korean, 43 for Mandarin, 32 for Japanese, 56 for Hindi, 36 for Spanish and 52 for German, were adopted in our experiments to construct the parallel phone recognizers. We used the IIR-LID [106] database to train the English and Japanese recognizer, the LDC Korean corpus (LDC2003S03) to train the Korean recognizer, the MAT corpus [97] to train the Mandarin recognizer, and the OGI-MLTS [94] (Multi-language Telephone Speech) database to train the other three recognizers. In the IIR-LID database, each language has more than 150 hours of speech, while in the OGI-MLTS database, German and Spanish has less than 1 hour of speech and Hindi has about 2 hours speech.

Phone recognizer	Training Corpus	Number of phones	Number of Gaussians
English	IIR-LID(150 hours)	44	32
Japanese	IIR-LID(150 hour)	32	32
Mandarin	MAT(70 hours)	43	32
Korean	LDC2003S03(25 hours)	37	32
Hindi	OGI-MLTS(2 hours)	56	6
Spanish	OGI-MLTS(1 hour)	36	6
German	OGI-MLTS(1 hour)	52	6
UPR-Merge	All of above	300	6/32
UPR-Train	OGI-MLTS,OGI 22	164	6
UPR-ASM	OGI-MLTS,OGI 22	256	6

Table 4.1: Phone recognizers used in the thesis

The feature used in these phone recognizers are 39-dimensional feature vectors consisting of 12 MFCCs and normalized energy, plus their first and second order time derivatives. Utterance-based cepstral mean subtraction was applied to the MFCC features to remove channel distortion. The HTK toolkit [76] was used to train a 3-state hidden Markov model (HMM) for each of the 300 phones. The HMM states of the English, Korean, Mandarin and Japanese models have 32 Gaussians each, while the states of the other languages have 6 Gaussians considering the availability of training data. Phone recognition was carried out using a fully connected null-grammar network of phones. The phone recognition accuracy of the English phone recognizer on TIMIT is 62.13%.

The second set of language specific phone recognizers, PPR2, consists of three phone recognizers based on long temporal context developed by Brno University of Technology (BUT) [20]. The three phone recognizers of Czech, Hungarian and Russian, are trained on the SpeechDat-E speech databases. The Czech phone recognizer is trained from 12 hours of speech from 1052 Czech speakers recorded over the Czech fixed telephone network. The Hungarian phone recognizer is trained from 10 hours of speech from 1000 speakers. And the Russian phone recognizer is trained from 18 hours of speech from 2500 speakers. Different from the conventional short time cepstral feature based phone recognizers, the feature extraction in these phone recognizers make use of long temporal context feature, known as TRAPs (temporal patterns) [72]. For each phone recognizer,

three neural networks (NN) are trained to produce the phoneme posterior probabilities. The PPR2 has 158 phones in total in which 45 are for Czech, 61 for Hungarian and 52 for Russian respectively.

4.3.2 Universal phone recognizers

A Universal Phone Recognizer (UPR) can be interpreted as a listener who possesses phonetic knowledge of many languages. In human perceptual experiments, listeners with multilingual background often perform better than who only knows a single language when asked to identify unfamiliar languages [6]. Ideally an UPR is trained with samples of all existing languages. However, in practice, an UPR is often trained with samples from a limited set of languages based on the assumption that common sounds are shared among languages. In this thesis, we study three UPRs derived with different training methods.

The first UPR is created by lumping together all the phone models from seven existing IIR language specific phone recognizers described in section 4.3.1. We refer to this UPR as UPR-Merge, as shown in the third last row of Table 4.1. The UPR-Merge makes use of the seven available language specific acoustic models. We can assume that it covers most of the sounds in various languages.

The second universal phone recognizer, referred to as UPR-Train, is initially trained from the 6 languages story data in OGI-MLTS corpus [77] which adopts Worldbet as the phonetic symbols. There are about 1 hour of phonetic transcribed data for each of the 6 languages. Around 300 phonetic symbols are presented in the 6 languages story corpus. Due to the limited number of samples for some phones, only 164 phones with high occurrence were selected as the phone set of the UPR-Train. Each phone model is a Hidden Markov model that has 6 Gaussian mixtures in each state. The story data of OGI 22 languages corpus [95] (consisting of telephone speech from 22 languages: Eastern Arabic, Cantonese, Czech, Farsi, French, German, Hindi, Hungarian, Japanese, Korean, Malay, Mandarin, Italian, Polish, Portuguese, Russian, Spanish, Swedish, Swahili, Tamil, Vietnamese, and English) were recognized with the initial models, and the phone recognition results are used to adapt the initial models to generate the UPR-Train acoustic models. There are 300 phone models in UPR-Train.

The third universal phone recognizer, referred to as UPR-ASM, was trained in a fully data-driven manner. In contrast to UPR-Train, UPR-ASM does not require the phonetic definition or transcription of the training data. It is based on the assumption that the sound characteristics of all spoken languages can be covered by a universal set of automatically derived acoustic units. Their corresponding models, called acoustic segment models (ASMs) [21]. The ASMs are trained in a self-organized and unsupervised manner. The training data from different languages are first decomposed into small segments, the similarity of these segments are compared and those segments that are close to each other are combined as an acoustic unit. Segments of an acoustic unit are used to train an HMM model to represent the characteristics of this unit. The trained models are used to decode the training speech. The HMMs which are similar are grouped, and their decoding results are combined to train new HMMs. The process runs several iterations until stability is achieved. The story data of both OGI-MLTS corpus and OGI 22-languages corpus were used for the UPR-ASM training. In total, there are 256 ASMs in UPR-ASM, an ASM is an HMM model with 6 Gaussian mixtures in each state.

4.4 Experiments

In this section, a series of experiments are conducted to evaluate the proposed TOPT construction and selection methods for language recognition on NIST LRE96, LRE03 and LRE07 test data. The details of the development and test data set are introduced in Appendix A. Specifically, the following experiments are performed:

- In Section 4.4.1, we validate the effectiveness of the two phone selection criteria for TOPT construction.
- In Section 4.4.2, we examine the effectiveness of the proposed TOPT selection strategy and compare the performance of using different numbers of TOPT for language recognition.
- In Section 4.4.3, we explore the possibility of using higher order n -grams with the selected TOPTs.

- In Section 4.4.4, we compare the language recognition performance of using TOPTs generated from a language specific phone recognizer and TOPTs generated from a universal phone recognizer.
- In Section 4.4.5, we compare the language recognition performance of using 1-best phone sequences with using lattices based phonotactic statistics derived from TOPT front-end.

In the following experiments, all the speech segments were first processed by an energy based voice activity detection [96] program to remove silences, they are then decoded by the phone recognizers or phone tokenizers (TOPTs) without any linguistic constraints, i.e. free phone loop grammar is used in the decoding process. All the language recognition experiments are conducted on a PPR-VSM system (as described in Section 3.2).

4.4.1 Phone selection

We experimentally validate our phone selection strategy for TOPT construction. Firstly, we show that different phones have different contributions on language separation. Secondly, we validate the effectiveness of the phone selection methods by conducting language recognition experiments on TOPTs derived from the proposed phone selection criteria.

The first experiment was conducted on the CallFriend corpus which includes 12 languages. Figure 4.2 shows the discriminative index of English phones in selecting Mandarin from other languages using separation margin (SM) and mutual information (MI) criteria as described in Section 4.2.1.1. Although SM and MI suggest different discriminative ability of phones, the two curves clearly demonstrate that the discriminative index of phones varies for a given target language. The result suggests that we may eliminate some low discriminative ability phones in language recognition system without losing much discrimination power. Hence it is reasonable to select a subset of phones with respect to a target language to form a new target-oriented phone tokenizer (TOPT).

The pie chart besides each curve shows the discriminative index of the four broad phone classes. The discriminative index of each phone in the same phone classes is averaged to derive the discriminative index of the class. In both criteria, the vowel class

Figure 4.2: Discriminative index of English phones for the target language Mandarin based on separation margin (SM) and mutual information (MI) criteria

has the highest discriminative index and the stop class has the lowest. This suggests that although the two criteria have different phone ranking, the discriminative index by phone class is similar. It reveals that the two proposed criteria are consistent on the discriminative ability of the broad phone classes. From the pie charts, we can also find that the group of vowels are the most important phone class for language recognition and the nasals are the least sensitive phones for language recognition.

The next experiment was devised to study the language performance of using different number of phones in each TOPT. If the number of phones in an TOPT is too small, the resulting TOPT may be unable to cover all the sounds and sound patterns in the target languages, hence the TOPT will not have sufficient discriminative characteristics to separate languages. On the other hand, if a large number of phones is used in each TOPT, redundant information is obtained in tokenization process.

Figure 4.3 shows the equal error rates (EER) of using 12 English TOPTs as PPR front-end on LRE03 30-second test set. The dotted line indicates the language recognition results using a single English phone recognizer with 44 phones as our baseline system. The two solid lines in Figure 4.3 show the language recognition performance when applying TOPTs derived from SM and MI phone selection criteria with different

Figure 4.3: Language recognition using 12 English TOPTs with different numbers of phones in each of the TOPTs on LRE03 30-second trials

numbers of phones. Bigram phonotactic features were used to generate the discriminative vectors in the VSM back-end in each of the 12 TOPTs. The dimensionality of the phonotactic feature vectors is dependent on the number of phones used in each TOPT. For example, when we have 20 phones in each TOPT, the total dimension of the bigram feature vector is 20×20 . If there are 40 phones in each TOPT, the feature dimension will be increased to 40×40 . In decision making, the output scores from each of the 12 TOPT front-ends were averaged to derive final scores. When the number of phones in each TOPT is greater than 17, the TOPTs outperform the original English phone recognizer. The experimental results also show that there is no significant improvement when putting more than 20 phones in each TOPT. Hence, we chose the top 20 most discriminative phones an TOPT for the following experiments.

Next, we studied the distribution of phone presence in all the TOPT candidates derived from the English phone recognizer. The TOPT of each target language contains 20 phones and the CallFriend 12 languages are used as target languages. Figure 4.4 shows the number of times a particular phone in the English phone inventory is selected for the 12 TOPTs. One observation from Figure 4.4 is that 17 phones were commonly selected in 50% of the TOPTs. Recall that there are only 20 phones in each TOPT, it means those TOPTs have highly overlapped phone set. Redundant information may obtained by using these TOPTs as front-end which increases the computational complexity of the

Figure 4.4: The number of times each phone in the English phone recognizer is selected in the 12 TOPTs

Tokenizers/EER(%)	LRE96	LRE03	LRE07
English(44 phones)	5.63	7.71	10.51
English(36 phones)	5.84	8.11	10.83

Table 4.2: Language recognition performances (EER%) of the original English phone recognizer (44 phones) and an English phone tokenizer with only 36 selected phones

language recognition system. This suggests that it is necessary to do selection among the TOPT candidates.

Another observation from Figure 4.4 is that only 36 out of 44 English phones were selected in these TOPTS, i.e., 8 phones have never been selected in any TOPT. We assume these 8 phones have little discriminative abilities compared to other phones. To validate this assumption, we performed an experiment using a phone tokenizer with 36 selected phones and compared its language recognition performance to the original English phone tokenizer which has 44 phones. Table 4.2 shows the language recognition performances (EER%) of the experiments on LRE96, LRE03 and LRE07 30-second data sets. From the results, we observe that the two tokenizers give very similar performance on all the three 30-second test sets. By removing 8 (18%) of less informative phones, the system resulted in only a small loss on language recognition accuracy (around 3% relatively). The result confirms that some phones are not as discriminative as other phones for language recognition. Of the 8 unselected phones, we noticed that 5 of them

are long vowels and the other 3 are long consonants. One explanation of why these phones are not selected may be that the phone recognition favours the short models, as a result of the null-grammar decoding.

4.4.2 TOPT selection

In the previous subsection, the experimental result (Figure 4.4) shows that the derived TOPTs have highly overlapping phone sets. To reduce the redundancy caused by using these TOPTs as the PPR front-end, we proposed to perform TOPT selection by their merit value 4.7. The objective of the following experiments is to validate the effectiveness of the proposed TOPT selection method for language recognition.

Figure 4.5: Merit, discriminative ability and distinctiveness of TOPTs derived from the English phone recognizer

We carried out an experiment to examine the relationship of the merit, discriminative ability and distinctiveness of TOPTs, as described in Section 4.2.2. Figure 4.5 shows the the merit value, the discriminative ability and the distinctiveness of 12 TOPTs derived from the English phone recognizer using the separation margin (SM) phone selection criterion. The merit value of an TOPT is derived from its discriminative ability and the distinctiveness as in Equation 4.7. The two dotted lines in Figure 4.5 indicate the discriminative ability and distinctiveness of each of the 12 TOPTs. The solid curve shows the merit values of those TOPTs. Note that, to facilitate the visualization, the scores of

EER(%)	LRE96		LRE03	
	Top5	Bottom5	Top5	Bottom5
TOPT-English	4.63	6.71	6.25	8.86
TOPT-Hindi	6.67	6.83	7.65	9.54
TOPT-Korean	5.21	6.97	7.29	8.55
TOPT-Mandarin	4.59	5.18	6.77	7.38
TOPT-Japanese	4.69	5.55	6.75	8.12
TOPT-German	5.95	8.55	8.18	11.05
TOPT-Spanish	5.60	6.29	7.59	8.80

Table 4.3: Performance comparison between Top5 and Bottom5

the discriminative ability, the distinctiveness, and the merit were normalized to the same range. Although some TOPTs have high discriminative abilities, their merit values are low as they have low distinctiveness. For example, the 10th TOPT in Figure 4.5.

To incorporate more discriminative information for language separation and to keep low redundancy, our strategy is to select TOPTs with the highest merit values to form the PPR-VSM front-end. We study the language performance with different number of TOPTs in the PPR front-end. The experiment is conducted using the 12 TOPT candidates derived from the English phone recognizer. There are 20 phones in each TOPT selected under the separation margin (SM) criterion. First we rank the TOPT candidates by their merit values as given in Equation 4.7, then we select TOPTs from the ranked candidate list to form the PPR front-end. The selected TOPTs are used in language recognition on the LRE96 and LRE03 30-second data sets.

Figure 4.6 shows the language recognition EERs obtained with different number of TOPTs as PPR front-end on the LRE96 and LRE03 30-second test trials. In the n -gram feature extraction, bigram phonotactic statistics were used. The experiment results show that the overall language recognition performance is improved with the increasing of the number of TOPTs, and the improvement saturates after 5 TOPTs are used. The results confirmed our assumption that adding redundancy information does not improve the accuracy of language recognition.

Next, we verify the effectiveness of the proposed tokenizer selection strategy by comparing the top 5 TOPTs (denoted as Top5) with the highest merits and bottom 5 TOPTs (denoted as Bottom5) with the lowest merits. Table 4.3 shows the EERs of Top5 and

Figure 4.6: Language recognition performance (EER%) as a function of the number of TOPTs

Bottom5 derived from 7 language specific phone recognizers (as described in Section 4.3) respectively. For all the 7 phone recognizers, the Top5 TOPTs consistently outperforms the Bottom5 in language recognition. The result shows the effectiveness of the proposed TOPT selection technique.

Another observation from Table 4.3 is that the German, Spanish and Hindi phone recognizer derived TOPT performs worse than the rest of phone recognizers. This can be explained by the fact that the three phone recognizers are trained from only 1 to 2 hours of speech data (as in Table 4.1) while the other phone recognizers are trained from 25 to 150 hours. The acoustic models of the three phone recognizers are therefore comparatively not well trained. This supports the finding in [20] that a phone recognizer which has high phone recognition accuracy also achieves better performance in language recognition.

In the following experiments, we continue to use the top 5 TOPTs that have the highest merit values to form the PPR front-end.

4.4.3 Bigram vs. trigram phonotactic statistics

In the PPR-VSM framework, we expect that higher order n -gram statistics contain more discernible phonotactic information. However, to use higher order n -gram, we will need

significantly more training samples as higher order n -gram results in higher dimensional n -gram statistic feature vectors. In practice, to overcome the limitation of the training data and computation capability, up to trigram phone statistics are usually used in a practical PPR-VSM systems [33, 86].

Figure 4.7: Performance comparison of bigram vs trigram phonotactic features on the LRE03 30-second trials

As an TOPT employs a smaller phone inventory, the use of higher order n -gram phonotactic statistics is feasible. For example, the English phone recognizer that we used in previous experiments has 44 phones in its phone inventory. If trigrams phone statistics are included in the feature vector, the upper bound of the feature vector dimension will be $44 + 44 \times 44 + 44 \times 44 \times 44 = 87,164$. While for an TOPT with 20 phones, the dimension upper bound of the trigram feature vector will be $20 + 20 \times 20 + 20 \times 20 \times 20 = 8,420$, which is much more manageable. If we use 5 TOPTs in the PPR front-end, we end up with a composite vector of $5 \times 8420 = 42,100$ in feature dimension, which is still more compact than that with single English phone recognizer. For language recognition, as not all possible n -grams are observed, the feature dimension is actually smaller than the upper bound. Experiments indicate that the average dimension of the trigram feature vectors of using 7 language specific phone recognizers is 85,420, and the average dimension of trigram composite feature vector with 5 TOPTs is 40,332.

We compare the language recognition performance using bigram and trigram phonotactic statistics. Figure 4.7 reports the EERs on the LRE03 30-second trials with different numbers of TOPTs using bigram and trigram statistics features. All the TOPTs were derived from the English phone recognizer and each TOPT had 20 phones. The two dotted lines show the performance of the system using only the English phone recognizer in the front-end with phonotactic information presented by bigram and trigram statistics respectively. The two solid lines represent the performance of the PPR-VSM systems using the top 5 TOPTs based on bigram and trigram statistics. It shows that the systems with trigram phone statistics consistently outperform those with bigram phone statistics. By using TOPTs instead of the original phone recognizer as the front-end tokenizer, the language recognition performance is improved without significantly increasing the dimension of the feature vectors.

4.4.4 Language specific phone recognizer vs. universal phone recognizer

In the previous experiments, we have validated the performance improvement of using target-oriented phone tokenizer derived from language specific phone recognizer on language recognition. In this section, we examine the TOPTs derived from the universal phone recognizer.

In human perceptual experiments, listeners with multilingual background often perform better than those who only know a single language when asked to identify unfamiliar languages [6]. A universal phone recognizer (UPR) can be thought of as a person who has knowledge of multiple languages, while a language specific phone recognizer can be thought of as a person who knows only one specific language. In this section, we first compare the performance of 3 UPRs and 7 language specific phone recognizers for language recognition. We then compare the language recognition performance of using TOPTs derived from UPRs and language specific phone recognizers.

Table 4.4 shows the EERs of language recognition systems on LRE96, LRE03 and LRE07 30-second data sets. We report the experimental results of using 7 language specific phone recognizers and 3 UPRs in the front-end with bigram phonotactic features. The UPRs are UPR-Merge, UPR-Train and UPR-ASM as described in Section 4.3. For

Phone Tokenizer	LRE96	LRE03	LRE07
English	5.63	7.71	9.32
Hindi	8.33	10.83	11.41
German	7.26	9.22	12.36
Korean	6.73	8.05	10.76
Spanish	8.93	10.93	13.96
Japanese	5.89	8.66	9.79
Mandarin	5.37	7.97	9.38
UPR-Merge	4.14	6.46	8.39
UPR-Train	4.41	6.55	8.52
UPR-ASM	4.90	6.64	9.74

Table 4.4: Language recognition performance with UPR and language specific phone recognizer as front-end tokenizers (EER%)

Phone Tokenizer	LRE96	LRE03	LRE07
TOPT-English(SM)	3.75 (33.39%)	5.04 (34.63%)	5.64 (39.49%)
TOPT-Hindi(SM)	4.52 (45.74%)	6.23 (42.47%)	7.74 (32.16%)
TOPT-German(SM)	4.25 (41.46%)	6.27 (31.99%)	8.20 (33.66%)
TOPT-Korean(SM)	4.40 (34.62%)	6.04 (24.97%)	7.13 (33.74%)
TOPT-Spanish(SM)	5.54 (37.96%)	8.06 (26.26%)	9.04 (35.24%)
TOPT-Japanese(SM)	3.66 (37.86%)	5.47 (36.84%)	6.36 (35.04%)
TOPT-Mandarin(SM)	2.85 (46.93%)	4.30 (46.05%)	5.68 (39.45%)
TOPT-UPR-Merge(SM)	2.20 (46.86%)	3.23 (50.00%)	4.64 (44.70%)
TOPT-UPR-Train(SM)	2.27 (48.53%)	3.53 (46.11%)	5.23 (38.62%)
TOPT-UPR-ASM(SM)	2.91 (40.61%)	3.48 (47.59%)	6.78 (30.39%)
Fusion	1.27	1.42	2.73

Table 4.5: Language recognition performance (EER%) with TOPTs derived from UPR and from language specific phone recognizers as front-end tokenizers. In brackets are the relative EER reductions of the TOPT front-ends over their original phone recognizers

all three test sets the 3 UPR front-ends outperform any of the language specific phone recognizer. This result complies with the observation in a perceptual study [6] where the exposure to multiple languages helps human subjects in language recognition.

Table 4.5 summarizes the language recognition results using tokenizers derived from 7 individual language specific phone recognizers and the UPR-Merge in the front-end. To establish a fair comparison, we had the same number of tokenizers in different front-ends, and the same number of phones in each of the tokenizers. In each TOPT, 20 phones were selected with SM criterion. Each front-end consists of the top 5 TOPTs selected according

Phone Tokenizer	LRE96	LRE03	LRE07
UPR-Merge	4.14	6.46	8.39
TOPT-UPR-Merge(SM)	2.20	3.23	4.64
TOPT-UPR-Merge(MI)	2.58	3.87	4.75
UPR-Train	4.41	6.55	8.52
TOPT-UPR-Train(SM)	2.27	3.53	5.23
TOPT-UPR-Train(MI)	2.46	3.70	5.36
UPR-ASM	4.90	6.64	9.74
TOPT-UPR-ASM(SM)	2.91	3.48	6.78
TOPT-UPR-ASM(MI)	2.97	3.56	7.01

Table 4.6: Language recognition performance with UPR and their derived TOPTs as front-end tokenizers (EER%)

to their merits as defined in Equation 4.7. As expected, the result shows that the target-oriented phone tokenizers derived from a universal phone recognizer (UPR) consistently outperforms those derived from any language specific phone recognizers. Fusing all TOPT systems together by averaging further improves the language recognition performance, as shown in the last row of Table 4.5. As different TOPTs enhance the discriminative ability of different languages, a fusion system can exploit the complementary information to improve the recognition accuracy.

Next, we compare the language recognition performance of 3 universal phone recognizers and their derived TOPTs on language recognition.

Table 4.6 compares the language recognition performance using single UPR and UPR derived TOPTs in the front-end for language recognition on the LRE96, LRE03 and LRE07 30-second test sets. The top 5 TOPTs based on their merits were selected where each TOPT contains the 20 most expressive phones. We report the TOPTs constructed by separation margin (SM) and mutual information (MI) phone selection criteria. The number of phones in the 3 UPR phone recognizers are 300 (UPR-Merge), 164 (UPR-Train) and 256 (UPR-ASM) respectively, therefore the dimension of the trigram phonotactic feature is very large, and computationally prohibitive in the experiments. As a result, the phonotactic features in the UPRs are presented with only unigram and bigram phone statistics. However, the UPR derived TOPTs have a smaller number of phones than the original UPRs, hence the phonotactic features can be presented with unigram, bigram and trigram phone statistics. While observing that UPR derived TOPTs with

upto trigram outperforms UPR with upto bigram, we understand that it may not be a fair comparison to compare directly bigram and trigram results. The main idea here is to validate that the TOPT construction method make it possible to incorporate higher order phonotactic information for the language recognition task.

It is encouraging to find out that the UPR derived TOPTs consistently outperform the original UPR in all of the three test sets. The two phone selection methods achieve comparable performance while the SM method works slightly better. We believe this is due to the fact that the SM method utilizes the discriminative information among languages, while the MI method only concerns the dependency between the target language and the phones.

4.4.5 1-best vs. lattice phonotactic statistics

In the PPR framework, the role of the acoustic front-end is to extract accurate and rich phonotactic statistics for language characterization. Therefore, the coverage of the phonotactic statistics and the quality of phone recognizer, or tokenizer, has direct impact on the performance of the resulting system. While the link between the quality of the acoustic model and system performance can be easily understood, the coverage of phonotactic statistics has also been shown empirically to influence the system performance [25]. It was reported that with the same phone recognizer, the system could improve performance significantly by utilizing the phonotactic statistics from the lattice instead of 1-best phone recognition results. The improvement is attributed to the richer information available in the lattice, which captures the intermediate statistics of phone hypotheses.

Knowing that the TOPT technique enhances the discriminative ability of an PPR front-end by using 1-best phonotactic statistics, we investigate how the TOPT technique interacts with lattice-based phonotactic statistics. For simplicity, we experimented on the best system in Table 4.6, the TOPT-UPR-Merge system. Table 4.7 compares the system performance of the lattice-based with 1-best phonotactic statistics derived from TOPT-UPR-Merge. To establish a fair comparison, both systems employed the same front-end tokenizers that consisted of the top 5 TOPTs selected from UPR-Merge. Each TOPT has 20 phones which are selected based on the SM criterion. The experimental

Phone Tokenizer	LRE96	LRE03	LRE07
TOPT-UPR-Merge(SM)	2.20	3.23	4.64
TOPT-UPR-Merge-Lattice(SM)	1.50	1.65	3.54

Table 4.7: Language recognition performance with TOPTs derived from UPR-Merge using lattice and one best phonotactic statistics (EER%)

results show that the lattice-based approach consistently outperforms the 1-best approach across all three data sets. This observation agrees with other reported results [25] in that phonotactic information derived from lattice contains more discriminative information for automatic language recognition.

To summarize, given an existing phone recognizer and a development database of target languages, the two-step target-oriented phone tokenizer construction process allows us to design a new PPR front-end for a language recognition system that has multiple target-oriented phone tokenizers.

4.5 Summary

In this chapter, we presented a target-oriented phone tokenizer (TOPT) approach to construct a parallel phone recognizer (PPR) front-end from existing phone recognizer without requiring new transcribed training data. We validated the proposed method on the NIST LRE96, LRE03 and LRE07 30-second test sets.

The experimental results show that the language recognition performance can be improved by using the tokenizer derived from the proposed TOPT method. By comparing the performance of TOPT derived from language specific phone recognizers and universal phone recognizers, we observed that an universal phone recognizer provides richer language separation knowledge, hence gives better language recognition performance. When incorporating lattice information in phonotactic feature extraction, the language recognition performance is further improved than using 1-best phone sequences in the phonotactic feature extraction.

The TOPT method chooses a subset of phones to construct a new phone tokenizer. In the next chapter, an alternative method to construct new phone tokenizer, called target-aware language models (TALM) is presented. The TALM method extends the

idea of constructing target oriented front-ends from an existing phone recognizer. If we see TOPT as making hard decisions when selecting the phone subset, the TALM method provides a way to make soft decision by assigning weights to each phone according to its contribution to language separation.

Chapter 5

Target-aware Language Models for Language Recognition

5.1 Introduction

In the previous chapter, we proposed a discriminative feature selection method that derives a set of target-oriented phone tokenizers (TOPT) for automatic language recognition, where we construct parallel target-oriented phone tokenizers from a phone recognizer using a subset of phones that are uniquely important to the target languages. The TOPT approach is validated with a series of experiments. The experimental results show that the TOPT approach improves language recognition performance tremendously without requiring for additional labeled training data and acoustic modeling. Despite its promising results, the TOPT method can be further improved due to the following two limitations:

- The TOPT approach uses only a subset of phones from the original phone inventory and hence it has limited phone coverage.
- As an TOPT makes hard decision on selecting a subset of phones with high discriminative ability, the number of phones may affect the performance.

In the decoding process, the TOPT is forced to make some approximation on certain sounds that are not covered by the selected phone set. Such approximation introduces errors for phone recognition which affects the accuracy of the language classifiers. Refining the idea of target-oriented phone tokenizer, this chapter proposes a target-aware language models (TALM) method to address the above two concerns.

As stated in section 1.1, the most important difference between languages is the vocabulary and syntax. With lexical and syntactic information, humans are able to make sense of what they hear in a target language. In automatic speech recognition, the language model plays an important role in constraining the acoustic hypothesis with language priors. However, to incorporate the lexical and syntactic information in automatic language recognition, humans rely on a large vocabulary speech recognizer for each target language [46]. It is resource and time consuming to have a large vocabulary speech recognizer for each of the target language. Sometimes it is not achievable due to lacking of enough labeled training data from certain language. In this chapter, we propose a data driven method to obtain language related information from the phone recognition results. Then we incorporate this information in the decoding process of the automatic language recognition.

We propose to derive parallel phone tokenizers with target-aware language models (TALM) from an existing phone recognizer. With TALM, there is no need to make a hard decision on the choice of phones in a target-oriented phone tokenizer, all the phones are included with a language model. Each language model is trained to reflect the discriminative ability of individual phones for a specific target language. If the TOPT method is seen as making a hard decision on the choice of phones, the TALM method can be interpreted as making a soft decision on the discriminative phone selection.

In an automatic speech recognition system, the acoustic model encodes sources of acoustic variability such as speaker, channel, phonetic etc, while the language model imposes the lexical constraints that can be learnt from the training data. In the context of phone recognition, a proper language model provides prior knowledge of certain phones and phonotactics. This is similar to a human listening test where the subject pays special attention to the phones and phonotactics of interest.

In language recognition with phonotactic features, a phone recognizer is used to generate phone sequences from speech data of all the target languages. It is a common practice that no language model (free phone loop or null-grammar) is used in the decoding process as the speech data decoded by the phone recognizer are usually not in the language that the phone recognizer is trained from.

The target-aware language models (TALMs) will be generated by constructing a set of phone language models, each dedicated to a target language. In the target-aware language models framework, we are interested in deriving multiple phone tokenizers from an existing phone recognizer. The derived tokenizers will share a common set of acoustic models but each tokenizer will have a unique target-aware language model. A target-aware language model emphasizes on those phones that have higher discriminative abilities on separating specific language from others, and they are learned from the phone sequences derived from the original phone recognizer.

5.2 Target-aware language models construction

Figure 5.1: Target-aware language models: TALM

Figure 5.1 shows a block diagram of the TALM construction process. The following four steps are involved in TALM construction process:

- 1). The training utterances in all the target languages are processed by an existing phone recognizer to generate a set of phone sequences, the null-grammar is used in the decoding;
- 2). Based on the phone sequences generated in 1), we measure the discriminative ability of each phone in separating one target language from the rest;

- 3). Derive a language model for each of the target language according to the phone discriminative ability obtained in 2);
- 4). Construct TALM phone tokenizers, each tokenizer has a target language specific language model obtained from the previous 3 steps.

Hence, for a task of M target languages, we generate a new language model LM_m ($m \in [1 \dots M]$) for each of the target languages according to the discriminative ability of phones in separating one target language from the rest. In this way, a set of M new TALM phone tokenizers can be obtained. The TALM phone tokenizers share the same set of acoustic models, but each of them has a unique target language specific language model that is constructed according to the discriminative ability of phones in that target language.

5.3 Discriminative ability of phones

There are many ways to measure the discriminative ability of phones in a phone recognizer. For the TALM construction, we adopt the SVM margin (SM) criterion described in 4.2.1.1 to derive phone discriminative ability. Based on the phone occurrence frequency of the training utterances for all the target languages, we construct a linear SVM hyperplane to separate a target language from the rest target languages. The phone discriminative ability is proportional to the contribution of each phone in SVM hyperplane.

5.4 Target-aware language models

Given a phone recognizer and a set of training corpus, the discriminative ability of each phone in the phone inventory is examined to derive target-oriented language models. For a phone recognizer with e phones, the discriminative ability of phones toward target language m can be denoted as:

$$\{d_{m,1}, d_{m,2}, \dots, d_{m,i}, \dots, d_{m,e}\} \quad (5.1)$$

where $d_{m,i}$ denotes the discriminative ability of phone i to distinguish language m from the rest languages. Based on the discriminative ability of the phones in target language m ,

we can derive a unigram language model by approximating the phone’s prior probability with its discriminative ability,

$$LM_m = \{p_{m,1}, p_{m,2}, \dots, p_{m,i}, \dots, p_{m,e}\} \quad (5.2)$$

where the prior probability for phone i is computed as the normalized discriminative ability over all the phones for language m :

$$p_{m,i} = \log(d_{m,i}^2 / \sum_{j=1}^e d_{m,j}^2) \quad (5.3)$$

The language model gives higher prior to those phones that have higher discriminative ability for the target language separation. During the decoding process, the language model works together with the acoustic model to constrain the phone occurrence in the output phone sequences.

In speech recognition, grammar scale factor [76] is used to balance the contributions of acoustic and language models in the decoding process. It is an important parameter that significantly affects the accuracy of the speech recognition [103]. In TALM, we use the grammar scale factor to regulate the contribution of the language-dependent language models during the phone recognition, with respect to the acoustic models.

The target-aware language models (TALM) method described above can be seen as a generalized version of the target-oriented phone tokenizer (TOPT) approach, where in TOPT, some phones are removed during phone recognition and the selected phones are given equal probabilities. However, as the TALM method uses the full phone set to perform multiple phone recognition, the dimension of the generated phonotactic feature is larger than the TOPT method and experimental results show that the execution time for recognition is about 2-3 times slower.

It should be noted that the language modeling in proposed TALM method is very different from the LM widely adopted in the PPR-LM [7, 8, 20] architecture. Firstly, the TALM applies language model during the tokenization process in the front-end, while the PPR-LM uses language models to derive statistics from the output of the tokenizer in the back-end. Secondly, the target-aware language models are estimated from the discriminative ability of phones, which are different from the traditional n -gram statistics which is derived from the phone occurrence probability distribution in training corpus.

5.5 Experiments on TALM

Inspired by the findings in human perceptual experiments that listeners with multilingual background often outperform monolingual listeners in identifying unfamiliar languages [6], in the previous chapter, we proposed the TOPT method to derive multiple phone tokenizers from a universal phone recognizer, which have shown superior performance to those derived from a language specific phone recognizer. In this chapter, we compare the performance of the TALM derived phone tokenizers from a universal phone recognizer and the TALM derived phone tokenizers from a language specific phone recognizer.

In this section, the TALM based phone tokenizer construction method is evaluated on the 30-second test trials of three NIST language recognition tasks. The performance of the following three different tokenization front-ends will be compared: the original phone recognizer, the TOPT and TALM derived phone tokenizers. Both language specific phone recognizer and universal phone recognizer will be studied as the original phone recognizer to derive TOPTs and TALMs.

The training, development and evaluation corpus are described in Appendix A and the phone recognizers used in the experiments are described in section 4.3. In the TALM decoding process, the grammar scale factor is set to 15 for all the tasks. This value is derived from the experiments on the development set.

5.5.1 Language recognition with TALM

In this section, a language specific phone recognizer and an UPR phone recognizer and their derived phone tokenizer front-ends are studied. They are English phone recognizer, English-TOPT, English-TALM, UPR-Merge, UPR-Merge-TOPT and UPR-Merge-TALM.

The English-TOPT consists of a set of 12 target-oriented phone tokenizers (TOPTs), each contains a subset of 20 phones that have the highest discriminative ability in separating one target language from the rest based on the statistics of the CallFriend training data. A null-grammar is applied in each of TOPT decoding process. The final language recognition output scores are obtained by taking average on the individual TOPT output scores.

Tokenizers/EER(%)	LRE96	LRE03	LRE07
English Phone Recognizer	5.63	7.71	9.32
English-TOPT	4.63	6.25	7.67
English-TALM	3.60	5.65	6.63

Table 5.1: EER (%) of language recognition systems on 30-second test trials, with English phone inventory

Tokenizers/EER(%)	LRE96	LRE03	LRE07
UPR-Merge Phone Recognizer	4.14	6.46	8.39
UPR-Merge-TOPT	2.16	3.14	4.52
UPR-Merge-TALM	1.34	1.91	3.34

Table 5.2: EER (%) of language recognition systems on 30-second test trials with merged phone inventory

The English-TALM consists of a set of 12 phone tokenizers, they share the same set of acoustic models and each have a target-aware language model that emphasizes on those phones that have higher discriminative abilities, as described in the section 5.1. The discriminative ability of a phone is estimated using CallFriend training set. Each TALM has the same 44 phones as in the English phone recognizer. The output scores are obtained by taking average on the individual TALM output scores.

Table 5.1 shows the equal error rates (EERs) of three tokenization front-ends on NIST LRE96, LRE03 and LRE07 30-second test trials. By using the TALM derived phone tokenizers as front-end tokenizer, we consistently get the best language recognition performance among the three different phone tokenizer front-ends.

Similar experiments are conducted with the UPR-Merge phone recognizer which has a merged phone inventory from seven languages, and its derived TOPT and TALM. Table 5.2 shows the equal error rates using the UPR-Merge phone recognizer and its derived UPR-Merge-TOPT and UPR-Merge-TALM phone tokenizers on NIST LRE96, LRE03 and LRE07 30-second test trials. The UPR-Merge-TOPT consists of 12 tokenizers and each has 50 phones that have the highest discriminative ability in separating one target language from others. The UPR-Merge-TALM consists of 12 tokenizers, each has 300 phones. The 12 TALM tokenizers share the same set of acoustic models of 300 phones, and they are different only in language models. The language recognition output scores of UPR-Merge-TOPT and UPR-Merge-TALM are obtained by taking average on

the output scores of individual tokenizers. Note that the UPR-Merge-TOPT results in Table 5.2 are slightly different from that in Table 4.6. This is due to the fact that the UPR-Merge-TOPT in Table 5.2 consists of 12 TOPTs, while the one in Table 4.6 consists of only top 5 TOPTs.

By constructing TOPT, which only uses a subset of high discriminative phones in the decoding process, the language recognition performance is consistently improved. This confirms the effectiveness of target-oriented phone selection approach. We also observe that the TALM which has soft phone selection decision outperforms the systems based on either the original phone recognizer or TOPT tokenizer. This suggests that the soft-decision outperforms the hard-decision phone selection through target-aware language models.

5.5.2 Higher order language models

The experimental results from the TALM system in the previous section are based on unigram language models. It is possible to generate higher order n -gram language models using the same process as described in Section 5.4. To generate higher order n -gram language models, we only need to include higher order n -gram phone frequency statistics in the process to estimate their discriminative ability. One problem of this method is that there are many unseen n -gram phone combinations in the phone sequences derived from phone recognition. During language model generation, we therefore assign the smallest discriminative ability value to these unseen n -gram phone combinations.

By incorporating higher order language models in the TALM decoding process, more detailed target language information is included in the phone recognition process. We expect to see an improved language recognition performance. Table 5.3 shows the unigram, bigram and trigram TALM results on NIST LRE96 LRE03 and LRE07 30-second test sets.

The experimental results support our hypothesis that a better language recognition performance can be obtained by using bigram language models instead of unigram language models. However the language recognition accuracy is not improved by replacing bigram language models with trigram language models. One explanation for this is that many trigram phones were unseen in the phone decoding result, hence the derived

Tokenizers/EER(%)	LRE96	LRE03	LRE07
UPR-Merge-TALM unigram	1.34	1.91	3.34
UPR-Merge-TALM bigram	1.07	1.71	3.06
UPR-Merge-TALM trigram	1.07	1.70	3.14

Table 5.3: EER (%) of language recognition systems on 30-second test trials with merged phone inventory using different levels of n -gram language models

trigram language models are not well trained. Therefore, the language modeling construction method needs to be refined in order to incorporate trigram or even higher order information in front-end tokenization to improve the language recognition performance.

5.5.3 Discriminative ability of phones from UPR-Merge

In this experiment, the discriminative abilities of phones from different languages are compared. With an universal phone recognizer as the original phone recognizer, we would like to find out the contribution of phones from different target languages in language separation. The prior probabilities of phones in UPR-Merge-TALM tokenizers are analyzed. Figure 5.2 shows the distribution of top 50 phones that have the highest discriminative ability for separating English from the other 11 CallFriend languages. The distribution is derived from the absolute number of phones selected in TALM. Note that not all the English phones are among the most informative phone sets in separating English from the rest of languages. Many phones from Mandarin and Japanese phone recognizer are among the top contributors. This can be explained from the following aspects:

- Human listeners can distinguish one language from others even if they only have limited knowledge of that language.
- Listeners with a multilingual background often perform better than monolingual listeners in language recognition.
- Phones from the English, Mandarin and Japanese phone recognizers are the top contributors for English oriented TALM. One key reason is that in these experiments, the English, Mandarin and Japanese phone models are trained with more training data (as shown in Table 4.1). As such their phone models are well trained

Figure 5.2: Distribution of the top 50 phones in UPR-Merge-TALM for English language recognition

and hence contributed to better language recognition accuracy. This suggests that the quality of the phone models plays an important role in language recognition.

5.5.4 TOPT vs TALM

Note that the TOPT and TALM tokenizers reported in Table 5.1 and Table 5.2 have different number of phones. The number of phones in each TOPTs used in Table 5.1 and Table 5.2 are 20 and 50, while the number of phones in each TALM shown in Table 5.1 and Table 5.2 are 44 and 300 respectively. It would be more convincing if we can compare the language recognition performance of TOPT and TALM approach when they are using the same number of phones.

To make a fair comparison, we keep the same number of phones in TOPT and TALM. To this end, we first select the phones using TOPT approach, then we train a TALM on the set of selected TOPT phones (noted as UPR-Merge-TOPT-TALM). Figure 5.3 shows the equal error rates (EERs) of language recognition results on LRE03 30-second evaluation data. The two curves show the performance of UPR-Merge-TOPT and UPR-Merge-TOPT-TALM with different number of phones in the phone inventory. For a given number of phones, both UPR-Merge-TOPT and UPR-Merge-TOPT-TALM are using the same phone inventory. The phone inventory consists of phones which have the highest discriminative ability in separating a specific target language from other languages. The UPR-Merge-TOPT consists of 12 phone tokenizers that use null-grammar while UPR-Merge-TOPT-TALM consists of 12 phone tokenizers that use target-aware

Figure 5.3: EER (%) of different number of phones in UPR-Merge-TOPT and UPR-Merge-TOPT-TALM on LRE03 30-second test

language models (unigram). The language recognition output scores are obtained by taking an average on the output scores of individual tokenizers.

From Figure 5.3, we find that the UPR-Merge-TOPT-TALM consistently outperforms UPR-Merge-TOPT with different numbers of phones in the phone inventory. When the number of phones increases to over 50, the performance of UPR-Merge-TOPT in language recognition degrades. This can be explained by the fact that the TOPT treats each phone with equal importance, with the increasing of the phone numbers, the individual TOPT tends to become similar, thus providing little complimentary information. In contrast, the UPR-Merged-TOPT-TALM, which benefits from the discriminative language models, is able to achieve a better accuracy in language recognition even when a large number of phones is presented. The results suggest that TALM is more robust than TOPT in phonotactic front-end construction for automatic language recognition.

5.6 Summary

In this chapter, a target-aware language models (TALM) approach is proposed to construct multiple phone tokenizers. The TALM method is validated on NIST LRE96, LRE03 and LRE07 30 seconds test sets. Comparing the TOPT and TALM derived from a language specific phone recognizer and an universal phone recognizer, the universal

phone recognizer derived tokenizers achieve better language recognition performance as they provides richer language separation knowledge. With the same number of phones in TOPT and TALM, better language recognition accuracy can be achieved by using TALM as the front-end tokenizer. However one disadvantage of the TALM method is that it uses a full phone set to perform multiple rounds of decoding and requires more execution time and computing resources. Therefore, it may not be suitable for an online language recognition system which requires real time system response.

We also validate the TALM approach by incorporating higher order n -gram language models, the experiment results show there is positive effect by replacing unigram language models with bigram language models. However the language recognition performance is not improved by using trigram language models. The result suggests that more detailed language information is beneficial for automatic language recognition while the proposed language modeling method needs to be improved to overcome the data sparseness issue.

It is observed that the performance of automatic language recognition greatly degrade with shorter test segments. In the next chapter, we will concentrate on language recognition on short length speech segments. We propose to use a relevance feedback technique to augment the short length test segment to capture more language specific information. The details will be provided in the next chapter.

Chapter 6

Language Recognition with Relevance Feedback

6.1 Introduction

Recent results show that the performance of an automatic language recognition system is influenced by the length of the test utterance. It is observed in the NIST LRE tasks that the equal error rates increase substantially with the length of the test utterance reduces from 30 seconds to 3 seconds [21]. In light of this, we are prompted to study ways to augment short test samples using the automatic relevant feedback technique by making use of the labeled training samples, with a focus on language recognition of 3-second test utterances in NIST LRE tasks.

Relevance feedback is one of the major research areas in information retrieval [99, 100]. The original idea behind the relevance feedback is to involve users in the retrieval process. The information retrieval system's performance is improved by learning from user's feedback on the previous retrieved results. In general, the relevance feedback is performed in the following steps:

- 1). The system returns a list of retrieved results to a query;
- 2). Users provide their feedbacks on the system retrieval results to indicate which of them are relevant to the original test query;
- 3). The original test query is refined based on the feedback given by users, the system performs another round of retrieval to generate a set of new retrieval results;

- 4). The step 2) and 3) repeated until no more changes to be made on the query.

Rocchio [101] proposed a way to incorporate the relevance feedback technique into vector space models for text document retrieval. In this method, a retrieval query is first represented as a vector of term occurrence statistics. The term occurrence statistics vector is used to derive the initial retrieval results. For each document in the retrieval results, users give their judgements on whether it is relevant or irrelevant. According to the input from users, a set of corresponding relevant and irrelevant reference vectors are generated, and they are used to refine the original query. A new query is formed by taking a weighted sum of the original term occurrence vector and the relevant reference vectors. This method has proved to be effective for improving the accuracy of the information retrieval.

Inspired by its success in information retrieval, we propose to use the relevance feedback technique in automatic language recognition. In this thesis, we denote a test segment in language recognition as a test query, and the process of refining the test query to a new query as query expansion. In text based information retrieval, relevance feedback depends on the user's interaction to indicate the relevant and irrelevant documents. The terms from the relevant documents are then used to enhance the query while those from irrelevant documents are suppressed. However, in the automatic language recognition scenario, human assisted relevance feedback is not feasible because we can not assume the existence of a human agent at run time. The challenge of applying relevance feedback for automatic language recognition is to perform query expansion automatically.

One possible way to automate the process is to use the similarity between a test query and a reference utterance as an indication of their relevance. In our work, we first compare the similarity of the test query to all utterances in the reference corpus. The top ranked reference utterances are assumed to be relevant and the bottom ranked utterances are seen as irrelevant to the original test utterance. The relevant and irrelevant utterances are then used to revise the n -gram statistics of the original test utterance to form a new test query. The query expansion in text information retrieval is a supervised relevance feedback, the proposed speech query expansion in this chapter can be seen as an unsupervised relevance feedback.

In the following sections, we will present three different methods to derive and make use of the relevant and irrelevant utterances for query expansion: query expansion with relevant utterances, query expansion with relevant clusters and query expansion with relevant and irrelevant utterances. The first two methods incorporate only the relevant utterances for query expansion and the third make use of both relevant and irrelevant utterances in query expansion.

6.2 Relevance feedback for language recognition

In information retrieval research, a relevance measure shows the extent that the topic of a retrieved result matches the topic of the query. The study in [102] shows that two documents have a high likelihood of being relevant to the information needed if they are similar to each other. Hence the similarity between two documents will reflect their relevance to a query. In this work, we denote an utterance as relevant to another if they are spoken in the same language. Hence, given a test utterance as a query, we can find some relevant utterances to improve the query. One way to find relevant utterances is to compare their similarity in terms of phonotactic characteristics to the original test utterance.

We define a held out data set from the development set as a reference corpus for the language recognition task. The reference corpus consists of speech utterances from each of the target languages. Given a test utterance, our objective is to derive the relevant and irrelevant reference sets for test query refinement. The relevance of the utterances in the reference corpus are measured by their similarity to the original test query.

In the PPR-VSM language recognition system, a speech segment is converted into a n -gram statistic vector by one or many tokenization front-ends. Given two n -gram statistic vectors $\mathbf{v}_a$ and $\mathbf{v}_b$, both of them have dimension of n . The cosine distance of the two vectors is used to measure their relevance:

$$d(\mathbf{v}_a, \mathbf{v}_b) = \frac{\mathbf{v}_a \cdot \mathbf{v}_b}{\|\mathbf{v}_a\| \|\mathbf{v}_b\|} \quad (6.1)$$

where $\|\mathbf{v}_a\|$ and $\|\mathbf{v}_b\|$ are the magnitude of vectors $\mathbf{v}_a$ and $\mathbf{v}_b$, hence:

$$\|\mathbf{v}_a\| = \sqrt{v_{a1}^2 + \dots + v_{an}^2} \text{ and } \|\mathbf{v}_b\| = \sqrt{v_{b1}^2 + \dots + v_{bn}^2} \quad (6.2)$$

In Equation 6.1 the cosine distance value of two vectors will be in the range from 0 to 1, where 0 indicates that the two speech segments are orthogonal and hence they are exactly irrelevant, while 1 means that the two speech segments are identical.

Under the PPR-VSM frame work, each utterance in the reference corpus and test test is converted into a n -gram vector by the phonotactic feature extraction process described in Section 3.2. We propose the following three different approaches to do query expansion on the n -gram vector of the test utterance:

- Query expansion with relevant utterances: the original test query is expanded by the relevant utterances from the reference corpus.
- Query expansion with relevant clusters: the utterances in the reference corpus are first classified into clusters, the original test query is expanded by the clusters that are the most relevant to test query.
- Query expansion with relevant and irrelevant utterances: both relevant and irrelevant utterances in the reference corpus are used to expand the original test query.

Note that in the first two methods, only the relevant utterances are used for query expansion while the third method uses both relevant and irrelevant utterances in the query expansion. The details of the three query expansion methods are explained in the following sections.

6.2.1 Query expansion with relevant utterances

Figure 6.1 shows the process of the query expansion with a set of relevant utterances. We use dotted, meshed, striped and grid blocks to represent reference vectors in different languages. Given an input test query and a reference corpus, the query expansion is performed in the following steps:

- 1). We calculate the similarity of the input test query with every utterance in the reference corpus. The similarity is calculated with cosine distance as specified in Equation 6.1;
- 2). We rank all the utterances in the reference corpus according to their similarity to the test query;

Figure 6.1: Speech query expansion with relevant utterances

- 3). We assume the top n utterances are relevant to the test query. The n -gram statistic vectors of these utterances are used to augment the input query utterance. An augmented test query is used as a new input query to the language recognition system.

In this way, the expanded query is a weighted sum of the original query and the top n most relevant vectors in the reference corpus:

$$\mathbf{v}_e = (1 - \beta_u)\mathbf{v}_o + \beta_u \sum_{i=1}^k \frac{R_i}{k} \quad (6.3)$$

where $\mathbf{v}_o$ is the n -gram statistic vector of the original query, $\mathbf{v}_e$ denotes n -gram statistic vector of the expanded query, k is the number of relevant utterances, R_i denotes the n -gram statistic vector of the k -th relevant utterance and β_u is the weighting parameter which can be obtained from training data. It is worth to mention that the top n most relevant utterances may not be from the same language.

6.2.2 Query expansion with relevant clusters

It is generally agreed that the n -gram statistics obtained from long utterances are more reliable than that from short utterances. We believe a cluster of similar utterances has

Figure 6.2: Speech query expansion with relevant clusters

a better representation on the common statistics of the language than an individual utterance. In the second query expansion method for language recognition, we propose to use clusters of the reference utterances instead of individual reference utterances.

Figure 6.2 illustrates the process of relevance feedback with reference clusters. In the reference corpus, the utterances of the same target language are organized into several clusters using the k -means clustering technique [88]. The mean vector of a cluster is used to represent the cluster, and it is used to estimate the similarity of the cluster to the test query. In Figure 6.2, different blocks: dotted, meshed, striped etc represent n -gram statistic vectors of different languages and the inner dotted boxes represent clusters.

Given a test query, the query expansion with reference clusters is performed using a similar process as in query expansion with reference utterance:

$$\mathbf{v}_e = (1 - \beta_c)\mathbf{v}_o + \beta_c \sum_{i=1}^k \left(\frac{1}{k} \sum_{j=1}^n \frac{R_{jk}}{n} \right) \quad (6.4)$$

where $\mathbf{v}_o$ is the n -gram statistic vector of the original query, $\mathbf{v}_e$ denotes n -gram statistic vector of the expanded query, k is the number of relevant clusters, j is the number of utterances in the cluster, R_{jk} denotes the j -th n -gram statistic vector in the k -th relevant clusters and β_c is the weighting parameter which can be obtained using training data. Similar to the relevance feedback with relevant utterances as described in previous approach, the top k clusters may or may not belong to the same language.

6.2.3 Query expansion with relevant and irrelevant utterances

In document retrieval with relevance feedback [101], Rocchio formulated a way to combine the initial retrieved documents with both the relevant and irrelevant documents. In his approach, the relevant terms are added to the original query and those irrelevant terms are removed from the query. Motivated by the success of this approach in document retrieval, we propose to use both the top and bottom ranked reference utterances retrieved by the input query as representative sets of relevant and irrelevant utterances. The input query is then augmented by the relevant utterances and refined by the irrelevant utterances. The query expansion with relevant and irrelevant utterances can be formulated as:

$$\mathbf{v_e} = (1 - \beta_r - \gamma)\mathbf{v_o} + \beta_r \sum_{i=1}^{n_1} \frac{R_i}{n_1} - \gamma \sum_{j=1}^{n_2} \frac{S_j}{n_2} \quad (6.5)$$

where $\mathbf{v_o}$ is the n -gram statistic vector of the original query, $\mathbf{v_e}$ denotes n -gram statistic vector of the expanded query, R_i denotes the n -gram statistic vector in i -th relevant utterance, S_j denotes n -gram statistic vector of the j -th irrelevant utterances, n_1 and n_2 are the numbers of relevant and irrelevant utterances respectively. β_r and γ are the parameters that control the contributions of relevant and irrelevant feedback utterances. This method is different from the previous two methods in the use of irrelevant reference utterances S and weighting parameter γ .

The proposed expansion by relevant and irrelevant utterances is illustrated in Figure 6.3. Given a test speech query, the query expansion is performed in the following steps:

- 1). We calculate the similarity of the input utterance with each of the reference utterance in the reference corpus;
- 2). We rank all the utterances in the reference corpus by their similarity to the input query;
- 3). We select top n_1 utterances as the set of relevant utterances R and bottom n_2 utterances in the reference corpus as set of irrelevant utterances S . The n -gram statistic of the relevant and irrelevant utterances are combined with the n -gram statistics of the input query utterance according to Equation 6.5.

Figure 6.3: Speech query expansion by relevant and irrelevant utterances

6.3 Experiments on query expansion

In this section, we validate the effectiveness of the proposed query expansion methods for automatic language recognition. The three test sets from NIST LRE96, LRE03 and LRE05 are used as test corpus.

Without loss of generality, we use single phone recognizer as the tokenization front-end in the PPR-VSM framework. The language recognition experiment is performed with the Hungarian phone recognizer developed by BUT [20] as a single tokenization front end. Some details of the Hungarian phone recognizer can be found in Section 4.3. Bigram phonotactic statistics are derived from the decoded phone sequences.

We prepare a reference corpus using the CallFriend development set. There are 12 target languages in the CallFriend set. Two different accents data are included for the English, Mandarin and Spanish. In the reference corpus construction, we treat different accent as different languages, hence we have data from 15 languages in the reference corpus. The reference corpus is firstly processed by an energy based voice activity detection program to remove silence and then they are segmented into small speech segments. We select 200 utterances from each language randomly as reference utterances, each reference utterance is about 30 seconds in length. Hence we obtain a reference corpus of 3,000 utterances for 15 languages/accents. The same reference corpus is used for all the

Method/EER(%)	30-second	10-second	3-second
LRE96	3.75	14.71	28.93
LRE03	4.08	15.22	31.25
LRE05	7.30	16.50	29.41

Table 6.1: Baseline system equal error rates (EER%) on NIST 1996, 2003, 2005 LRE tasks with Hungarian phone recognizer.

experiments reported in this chapter. The left over data of the CallFriend development set are used as development set to determine the query expansion parameters(β_u , β_c , β_r and γ).

6.3.1 Baseline system

Firstly we conduct experiments with a standard PPR-VSM approach as a baseline, in which, the bigram statistics are derived from the phone sequences of the test speech segments, and they are used for language recognition. In the baseline experiment, no query expansion is applied. The BUT Hungarian phone recognizer is used as a single tokenization front end. The language recognition performance of the baseline system on the NIST LRE96, LRE03 and LRE05 tasks are reported in Table 6.1.

The experimental results show that the language recognition accuracy degrades significantly with the decrease in the length of the test utterance.

6.3.2 Query expansion by relevant utterances

For query expansion by relevant utterances, as described in Section 6.2.1, we consider the top 40 utterances from the reference corpus as the relevant utterances. The relevant utterances are used to augment the initial test queries. The language recognition experiment results are reported in Table 6.2. In these experiments, the weighting parameter is set to $\beta_u = 0.2$ (as described in Equation 6.2.1). The value is derived from the development set.

To measure the significance of the query expansion on language recognition performance, we performed a T-test [107] on EER values obtained from the baseline and query expansion system. For each data length, the LRE96, LRE03 and LRE05 data are randomly separated into 15, 12 and 35 non-overlapped partitions respectively. Each partition

Method/EER(%)	30-second	10-second	3-second
LRE96	4.07	13.61	27.54
LRE03	3.95	15.06	30.21
LRE05	7.42	16.48	28.95

Table 6.2: Equal error rates (EER%) of query expansion with relevant utterances on NIST 1996, 2003, 2005 LRE tasks with Hungarian phone recognizer.

consists of approximately 100 test utterances. For each partition, a pair of EER values can be obtained by using baseline system and query expansion. The EER pairs from all the partitions of the same length are used to perform a paired, one tailed T-test. The value of T-test values are 4.26E-04, 8.56E-06 and 1.35E-08 for 30-second, 10-second and 3-second test set, respectively. The significance test results show that the reported results are significant at the critical value of 0.05.

Comparing Table 6.1 and Table 6.2, we find that 10-second and 3-second results have been improved. However, the EERs are only improved marginally and the query expansion has even caused a small accuracy drop on the longer (30-second) test queries. The reason for such degradation may be due to the following two reasons: firstly, the relevant utterances contain variations that are not related to the target language; secondly, despite some gain from the relevant information, some irrelevant information is also present during the expansion of the original test queries. In the next sections, we will evaluate the methods to refine the relevant information and remove the irrelevant information.

6.3.3 Query expansion by relevant clusters

In the experiments of query expansion by relevant clusters, as described in Section 6.2.2, we categorize the reference utterances from the same languages into 10 clusters using k -means clustering algorithm. With 200 reference utterances from each of the target language, we can get 10×15 clusters for the 15 language/dialects reference corpus. The n -gram statistic vectors in each cluster are averaged to derive an n -gram vector to represent that cluster, which is denoted as a cluster vector.

Given a n -gram statistic vector derived from the test utterance, we calculate its cosine similarity to each of the cluster vector. The top 4 cluster vectors are selected as relevant vectors. All the n -gram vectors in the relevant clusters are used to augment the test

Method/EER(%)	30-second	10-second	3-second
LRE96	4.03	13.55	27.39
LRE03	3.97	15.16	30.06
LRE05	7.38	16.43	28.45

Table 6.3: Equal error rates (EER%) of query expansion with relevant clusters on NIST 1996, 2003, 2005 LRE tasks with Hungarian phone recognizer.

query. The optimal weighting parameter $\beta_c = 0.2$ is obtained from development set. The language recognition performance on the LRE96, LRE03 and LRE05 tasks are reported in table 6.3. The same significance test as described in Section 6.3.2 is performed on the results. The value of T-test values are 3.65E-04, 2.15E-06 and 5.18E-08 for 30-second, 10-second and 3-second test set, respectively. The significance test results show that the reported results are significant at the critical value of 0.05.

Compared to the previous method of query expansion with relevant utterances, the language recognition performances on 3-second data sets are improved by using expansion with relevant clusters. There is only small improvement on the 10-second data set. A relevant cluster emphasize the similarity of the utterances in the same language, hence better language recognition accuracy is achieved compared to the relevant utterance method.

6.3.4 Query expansion by relevant and non-relevant utterances

In the experiments on query expansion by relevant and non-relevant utterances as described in Section 6.2.3, we follow the same process as that of query expansion with relevant utterances in ranking the relevance of the utterances. For each test query, we pick the top 20 utterances in the ranked list as relevant utterances and the bottom 20 utterances as irrelevant utterances. We then combine them with the original query vector as in Equation 6.5, the parameters set as $\beta_r = 0.35$ and $\gamma = 0.15$. The parameter values are derived from the development set. To measure the significance of the results, we also conducted significance test as described in Section 6.3.2. The value of T-test values are 2.34E-04, 3.35E-06 and 4.96E-08 for 30-second, 10-second and 3-second test sets, respectively. The significance test results show that the reported results are significant at critical value of 0.05.

Method/EER(%)	30-second	10-second	3-second
LRE96	3.97	13.54	26.81
LRE03	4.02	15.09	28.67
LRE05	7.28	16.40	27.94

Table 6.4: Equal error rates (EER%) of query expansion with relevant and irrelevant utterances on NIST 1996, 2003, 2005 LRE tasks with Hungarian phone recognizer.

Table 6.4 shows the language recognition results on the NIST LRE96, LRE03 and LRE05 data sets. Compared with the previous methods, the language recognition accuracy for the 3-second data sets is improved across the board using query expansion with relevant and non-relevant utterances. For the 30-second sets, query expansion has small improvements on the LRE03 and LRE05 test sets but negative results on the LRE96 test set. The results show that the language recognition accuracy is improved by removing some irrelevant information in query expansion. The query expansion method is effective on short test utterances and otherwise on long test utterances. It would be an interesting future work to examine how to effectively expand the queries.

6.3.5 Conclusion

From the above experimental results, we find that the proposed query expansion methods are effective on the performance improvement of the short length test segments. To further validate the experiments on short length test segments, we conduct the language recognition experiments using a set of multiple phone recognizers as front end: PPR1 (as described in Section 4.3). The PPR1 consists of seven language specific phone recognizers developed by the Institute for Infocomm Research. The trigram phonotactic statistics are used in the experiments. Table 6.5 summarizes the language recognition performance of the baseline PPR1 system and the results with the proposed three query expansion methods on the 3-second data set.

By applying query expansion on the original test queries, the language recognition performance improved on all the three test sets. The results suggest that query expansion by relevant and non-relevant utterances has the best performance among the three approaches, with an EER reduction of 16.26%, 4.94% and 10.15% relative to the baseline results on the LRE96, LRE03 and LRE05 3-second tasks respectively.

Method/EER(%)	LRE96	LRE03	LRE05
PPR1 Baseline	21.16	21.25	24.23
PPR1 by relevant utterances	19.35 (8.55%)	20.54(3.34%)	22.77(6%)
PPR1 by relevant clusters	17.91(15.35%)	20.12(5.32%)	22.18(8.46%)
PPR1 by relevant non-relevant utts	17.72(16.26%)	20.20(4.94%)	21.77(10.15%)

Table 6.5: Equal error rates (EER%) for 3-second test queries of NIST 1996, 2003, 2005 LRE tasks with PPR1 phone recognizers. The percentages in the brackets are the relative improvement compared to baseline.

6.4 Summary

In this chapter, we have studied three ways to apply the relevance feedback technique for spoken language recognition. Assuming we have a labeled multilingual corpus, we exploit the way to use the reference corpus to automatically augment the test query. Three different approaches on query expansion are studied in this chapter.

Experimental results show that the proposed methods are especially effective for test queries in short length. The unsupervised relevance feedback method reduces the relative equal error rate by 16.26%, 4.94% and 10.15% on the NIST LRE96, LRE03 and LRE05 databases respectively for 3-second trials.

Query expansion provides a possibility to improve short length utterance language recognition performance at a low cost. However the quality of the reference corpus may affect the performance of the query expansion. It is important to ensure the reference corpus is close to the test task in terms of acoustic and channel conditions. The experimental results in Table 6.5 show that query expansion methods provide more improvements on the LRE96 test sets. It is evidenced that a matched reference corpus is beneficial for query expansion as both the LRE96 test set and the reference are extracted from the CallFriend corpus.

In this chapter, we discussed query expansion in the context of phonotactic based language recognition system. Without loss of generality, the query expansion methods can be extended to any other common language recognition framework, such as acoustic GMM, spectral SVM, with only minor modifications to the query expansion process. In the next chapter, we would like to discuss the way to reduce the feature dimension without affecting the language recognition performance.

Chapter 7

Phonotactic Feature Selection for Language Recognition

7.1 Introduction

In the previous chapters, we have discussed some techniques to enhance the discriminative ability of phonotactic features for language recognition. Firstly, parallel phone recognizers are adopted as the decoding front-end [32, 33] to enrich the phonotactic statistics over a single phone recognizer. Secondly, the phonotactic features derived from lattice instead of the top one decoding phone sequences contribute to improved language recognition performance [25]. This improvement is attributed to the rich information available in the lattice (Section 4.4.5). Lastly, we have shown that features that include higher order n -gram statistics contain more discernible phonotactic information and will achieve better language recognition performance [20, 33] (Section 4.4.3). While these techniques improve the performance of language recognition, the dimensionality of the feature vector is increased significantly. Modeling phonotactic features in a high dimensional feature space will lead to higher computational cost and it requires a large amount of training data which may not be available. In this chapter, we explore feature selection techniques on n -gram statistic features. Our objective is to reduce the dimensionality of the feature while maintaining their discriminative characteristics.

Feature selection is a strategy to keep only important features by removing features that are less informative. Various feature selection methods can be grouped into two broad groups: the filter group and the wrapper group [98]. Wrapper approaches use a

search algorithm to search through the feature space and train a model for each subset, the model is then evaluated by its classification accuracy on test data. The feature subset which gives best accuracy is selected. Wrapper methods can be computationally expensive and the trained model is prone to over-fit to the data set. Filter methods are similar to wrapper methods in the search approach, but instead of evaluating the selection against a model, the filter is stopped when the selected subset satisfies a predefined condition.

In this chapter, we study two criteria for phonotactic feature selection, specifically, separation margin and Chi-squared test [37]. The separation margin method can be categorized into the wrapper type of feature selection, it uses a set of SVM models trained from the original feature space to derive the merit of each feature. The Chi-squared test measure is a filter type of feature selection, it measures the dependency of a feature and target languages as the merit value. Both methods select phonotactic features according to their contribution for language separation. Experimental results show that the proposed feature selection methods reduce the dimensionality of the phonotactic feature vectors while maintaining the language recognition performance.

7.2 Phonotactic feature selection

In the PPR-VSM framework, the dimension d of the composite phonotactic feature vector is equals to the total number of possible n -gram patterns:

$$d = \sum_{i=1}^F \sum_{j=1}^k (e_i)^j \quad (7.1)$$

where F is the number of the phone recognizers, e_i denotes the number of phones in the i -th phone recognizer and k denotes the order of n -gram considered. Equation 7.1 shows that the feature dimension d increases with the number of phone recognizers F or the order of n -gram k . Modeling high dimensional vectors requires large amount of training data and long computational time, which may not be feasible for an online language recognition system which requires real time system response. Hence, it is desirable to introduce a feature selection technique to reduce the dimensionality of the feature vector.

In this section, we first introduce some methods which are commonly adopted in phonotactic feature selection. We then propose two criteria to measure the merit of features for automatic language recognition. A feature element with high merit value indicates that this feature element is important for language recognition. In the first method, the n -gram statistics features are measured by their contribution to the language separation margin in a support vector machine. In the second method, the Chi-squared value of the feature element and target languages is used to measure the importance of features.

7.2.1 Previous works

In this section, three methods for selecting phonotactic features are introduced, specifically: selection by occurrence counts [36], principal components analysis [90] and iterative SVM [36]. The first two methods can be categorized to filter type of feature selection, as they perform feature selection based on the analysis of the training data. In these two methods, no language class information is used. The last method, the iterative SVM method is a combination of the filter and wrapper method and it makes use of the language information in the selection process.

7.2.1.1 Selection by occurrence counts

A common practice in the selection of phonotactic features is to eliminate phone n -grams with low occurrences. Given a set of training vectors with dimension d , where d is derived by Equation 7.1. The occurrence counts selection method eliminates features with occurrence counts less than a predefined threshold in the training corpus, specifically,

$$c_i < \theta \quad i \in [1, d] \tag{7.2}$$

where c_i is the number of times feature i observed in the training set and θ is the threshold. This method is straightforward but does not take into account the discriminative capability of phonotactic features. Thus the threshold θ is often set conservatively to a small value and only relatively small amount of features can be removed [36].

7.2.1.2 Iterative SVM

An iterative feature elimination method that utilizes both filter and wrapper methods was proposed in [36] to select discriminative features for language recognition. The following processing steps are performed iteratively:

- 1). A support vector machine (SVM) is trained for each of the target language using the training data in the original feature space;
- 2). Each feature in the original feature space is ranked by their contribution to the SVM separation margin;
- 3). Those low ranking features are eliminated from the feature space.

The remaining features are expanded to higher order n -grams, and the process repeats until a predefined condition is satisfied. With this method, a small subset of n -grams (referred to as keywords) have been shown to be sufficient for the language recognition task.

7.2.1.3 Principal component analysis (PCA)

Principal components analysis (PCA) is a classic technique used to project the input data from its original space into a new space with the aim to maximize the variance of the data set in the projected space.

Suppose we have a $n \times d$ data matrix X which consists of n rows of data samples with d features. Assume that X is mean centered, i.e., the estimated mean has been subtracted from each column. The PCA transformation is given by:

$$Y^T = AX \tag{7.3}$$

where A is an orthonormal matrix, each row a_i ($a_i \in 1 \times n$) is an eigenvector of X . In order to reduce the dimension of the original data, we use only the first t singular vectors of A to transform matrix X as defined in Equation 7.3. In this way, the original $n \times d$ dimensional matrix X is transformed into $n \times t$ dimensional matrix Y ($t < d$).

The PCA technique is widely used in data analysis and feature dimension reduction. It has also been used to reduce the dimensionality of a phonotactic based language

recognition system [90]. Although the PCA method finds subspaces that are useful for representing the original high dimensional vector space, the class information is not exploited in the projection process. Hence there is no reason to assume that the resulting projections will be useful for discrimination among data in different classes. Linear discriminant analysis (LDA) is another popular method in dimension reduction which makes use of class information [88]. It finds the optimal projection by maximizing the between-class distance and minimizing the within-class distance. However, when handling a very high dimensional data as phonotactic features, the traditional LDA algorithm may face the singular matrices problem when the number of training samples is less than the dimension of the feature vector. Hence the LDA method is difficult to directly apply on language recognition with phonotactic features, as the feature dimension is usually larger than the number of training samples.

7.2.2 Feature selection with separation margin (SM) and Chi-squared measure (CM) criteria

7.2.2.1 Separation margin (SM)

To derive the discriminative power of each features, we adopt the separation margin criteria as described in Section 4.2.1.1. For each target language, an SVM is trained using the n -gram composite vectors derived from the target language as the positive set and the feature vectors derived from all the rest of target languages as the negative set. An SVM [92] classifier learns a binary decision over a set of positive and negative feature vectors. In the SVM learning process, the optimization is performed by maximizing the separation margin between the positive set and the negative set. An SVM model is represented by a set of support vectors which define the separation hyper plane of the positive and negative sets.

As the weight for each feature element represents the contribution of the feature to the SVM separation margin, we use it to derive the merit of a feature element for language recognition. Assume there are M target languages in a language recognition task, given a feature element i , which presents a n -gram statistics, its weight in the j -th SVM is denoted as α_{ij} . The average weight to each of the M target languages is calculated as

	l	not l
t exists	A	B
t not exists	C	D

Table 7.1: Contingency table for a feature and a target language

the merit value of this feature for all the target language separation.

$$SM_i = \frac{1}{M} \sum_{j=1}^M \alpha_{ij} \quad (7.4)$$

This method is similar to the iterative SVM method discussed in Section 7.2.1.2 in ranking the discriminative ability of the features. However we eliminate the low ranking features using a different approach, which will be introduced in the following section.

7.2.2.2 Chi-squared measure (CM)

The Chi-squared statistic measures the lack of independence between a feature and a target language. The Chi-squared statistics of a feature i and a language l is calculated by using the following two-way contingency table [89]:

According to Table 7.1, the Chi-squared value of a feature t and language l is calculated as:

$$\chi^2(i, l) = \frac{N(AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)} \quad (7.5)$$

Here

- A stands for the number of times feature i occurs in training vectors of language l
- B is the number of times the feature i occurs in languages other than l
- C is the number of times other features occur in l
- D is the number of times features other than i occur in languages other than l
- N is the total number of training vectors

As defined in Equation 7.5, a higher χ^2 value means the feature i and language l is more dependent, and a value close to zero means the feature and language are independent.

Similar to the separation margin criterion we proposed in Section 7.2.2.1, the average Chi-squared value among all the target languages is used as the merit value of this feature element for language separation.

$$CM_i = \frac{1}{M} \sum_{l=1}^M \chi^2(i, l) \quad (7.6)$$

7.2.3 Efficient feature selection with relative mean difference (MD)

According to the two measures proposed in previous section, the n -gram features can therefore be ranked by their merit values. Given a ranked feature list, the dimension reduction algorithm is to find a suitable threshold to remove those less informative features without affecting the performance of the language recognition. In this section, we present an efficient method to find the feature selection threshold.

Figure 7.1: Features ranked by contribution to separation margin (merit values in $\log_e()$ scale)

As an example, Figure 7.1 shows the merit of features ranked by the separation margin (SM) criterion using trigram phonotactic features (details on the experiment setup will be described in the next section). For better illustration in Figure 7.1, the merit values

are presented in natural logarithm scale. The figure shows that the merit values decrease significantly in the top ranked features. This indicates that features at the top of the ranked list are more important as compared to the others. While in the middle portion of the curve, the decreasing slope of the merit values is small, this indicates that the features in this portion have comparable contributions for language separation. Removing some features in this portion might not affect the performance of the language recognition.

Agarwal et. al [91] proposed to use a difference of means method to detect change points from time series data of an enterprise system. A change point is triggered by a potential malfunction in the system. In their work, given any point under consideration, the mean difference between its left n and right n points is calculated. If the difference is greater than a predefined threshold, the change point is declared. A mathematical proof of the method can be found in the paper [91].

In order to have efficient feature selection from ranked merits, a variation of the mean difference method is proposed. A relative mean difference (MD) is used to set the feature selection threshold. Instead of finding the greatest difference of means, we find the smallest mean difference point for feature selection. Given a ranked list, the arithmetic left and right mean, $\mu_{i,l}$ and $\mu_{i,r}$ of any point i can be calculated by:

$$\mu_{i,l} = \frac{1}{n} \sum_{j=1}^n m_{i-j} \quad (7.7)$$

and

$$\mu_{i,r} = \frac{1}{n} \sum_{j=1}^n m_{i+j} \quad (7.8)$$

Where n denotes the number of left and right neighbors, and m denotes the sorted merit value. We define the mean difference (MD) of the point i as:

$$MD_i = \frac{\mu_{i,l}}{\mu_{i,r}} \quad (7.9)$$

As the features are sorted in descending order, we have $MD_i \geq 1$. A small MD value suggests there are small merit differences among the features in this $2n$ window area. We assume features in this window have similar contribution to the language separation. Hence, excluding some features in this small MD area may not affect the language separation performance. In phonotactic feature selection, for each point on the ranked list, we measure its mean difference value and we set the point with the minimum MD value as the feature selection decision point.

7.3 Experiments

7.3.1 Experiment setup

We have presented two feature selection methods based on different discriminative measurement criteria. In this section, we validate the proposed feature selection methods for the language recognition task. In our experiments, the PPR-VSM system architecture is adopted. Without loss in generality, we use a single phone recognizer, the Hungarian phone recognizer [20] developed in Brno University of Technology (BUT), as the front-end tokenizer to convert the input speech signals into phone sequences. In all the experiments, the phone lattices are obtained in the decoding process and used to derive n -gram statistics features.

We conducted the experiments on the test trials of the 2007 and 2003 NIST Language Recognition Evaluation (LRE 07, LRE03) tasks in which 14 target languages are involved in LRE07 and 12 target languages in LRE03. The language recognition system for LRE07 is trained on the LDC CallFriend corpus, the story data of OGI 22-language corpus [95], OHSU 2005 and LRE 2007 development data sets released by LDC . The language recognition system for LRE03 is trained on the LDC CallFriend data only. The language recognition performances are reported in terms of equal error rate (EER).

We proved that a higher order n -gram statistics may improve the performance of the language recognition in Section 4.4.3. Unfortunately, we are unable to try out the phonotactic information that is higher than trigram directly as the dimension of the feature vectors is beyond what our SVM tool can handle. By performing feature selection, the feature dimension is reduced, we can then recruit some higher order n -gram statistics into the feature vector. We assume the performance of the language recognition will be improved with the higher order phonotactic statistics. We will validate this assumption in the last subsection.

7.3.2 Comparison of different feature selection criteria

In the first experiment, we compare the language recognition performance on different number of features derived from four different feature selection methods. The first method is feature selection by occurrence counts. For each feature, the average n -gram occurrence

for each of the target languages is used as the merit of this feature element. Those features with low n -gram occurrence are eliminated from the feature vectors. The second method is feature selection by principal component analysis (PCA) as described in 7.3. In our experiments, the Matlab implementation of the PCA algorithm is used. As the PCA program has limitation on the dimension of the data, we removed some low counts features and kept only about 25k features. As LRE07 has 14 target languages, we selected 600 utterances from each of the target languages to form a 8400×25000 matrix for the eigenvector calculation. The other two methods are feature selection using SVM margin and Chi-squared criteria, as described in previous section.

As the Hungarian phone recognizer has 62 phones, the maximum number of n -gram element for unigram, bigram and trigram is: $d = 62 \times 62 \times 62 + 62 \times 62 + 62$ as described in Equation 7.1. In the experiments, not all the phone combinations can be observed in the decoding process, hence the actual number of n -gram patterns will be smaller. For the NIST LRE training data, there are 88292 features in the LRE07 training sets.

Figure 7.2: EER (%) with different numbers of features selected by different selection criteria on the LRE07 30-second test set

Figure 7.2 compares the language recognition performance of the features selected with four different feature selection methods on the LRE07 30-second data set. From

the results we found that the PCA method has the best performance for selecting small number of features, but it has difficulty in handling large number of features. As no class information is incorporated in the eigenvector calculation, the PCA method may not be able to find the optimal set of features that is the most discriminative for language separation.

The two proposed feature selection criteria: separation margin and Chi-squared value consistently outperform the occurrence counts method on the LRE07 30-second test set. The SVM-margin performs the best for 5k, 10k and 20k features, and all the methods converge at 50k features. Note that in Figure 7.2, the last point (80k) is the performance of language recognition without any feature selection (baseline). The two proposed methods achieve similar performance as the baseline while using only half of the features. The results suggest that the phonotactic feature is highly redundant, many features can be safely eliminated without significantly affecting the language recognition performance.

7.3.3 Validation on the feature selection with relative mean difference (MD)

In this subsection, we evaluate our proposed relative mean difference (MD) method for threshold selection. We rank the features using the two feature selection criteria proposed earlier and conduct a series of language recognition experiments with different numbers of selected features. In feature extraction process, up to trigram phone statistics are used, all the experiments are performed on the LRE 2007 30-second test set.

Figure 7.3 shows the relative mean difference (MD) of features ranked by the separation margin criterion on the LRE07 30-second test set. The relative mean difference values are derived using $n = 100$ as in Equation 7.9. The minimum MD point is found to be around 40000 features. Figure 7.4 shows the language recognition performance with different number of features selected by separation margin criterion on the LRE07 30-second test set. Comparing Figure 7.3 and 7.4 we note that the number of features for the best EER performance matches the feature selection based on MD.

Similar findings using the Chi-squared feature selection method are illustrated in Figure 7.5 which shows the MD value of features ranked with Chi-squared criterion on the LRE07 30-second test set. Figure 7.6 shows the language recognition performance with

Figure 7.3: Mean difference of ranked features derived by separation margin criterion on the LRE07 30-second set

different numbers of features selected by Chi-squared value on the LRE07 30-second test set. The minimum point of MD value is around 37000 features, and using corresponding top 37k features selected by Chi-squared merit gives best EER.

The experiment results show that the proposed threshold selection method based on mean difference value is effective to eliminate redundant features without affecting the performance on language recognition.

7.3.4 Combining two criteria

The two feature selection criteria, the separation margin and Chi-squared, measure the merit of a feature for language recognition differently. The Chi-squared method measures the dependency between an individual feature and the languages, while the separation margin method measures the contribution of a set of features to the language separation as a whole. In this section, we study the contribution of the two criteria to select phonotactic features for language recognition.

Given the two merits derived from separation margin (SM) as Equation 7.4 and Chi-squared measure criteria (CM) as defined in Equation 7.6, we propose a simple combined

Figure 7.4: EER (%) with different number of features selected by separation margin criterion on LRE07 30-second set

merit by taking a weighted average on the two merit values:

$$AM_i = \alpha_0 SM_i + \alpha_1 CM_i \quad (7.10)$$

where α_0 and α_1 are weighting parameter derived from the training set. In this way, all the features are ranked by the combined merit. We compare the language recognition performance of these three criteria by taking same number of features from the ranked lists derived from the three criteria. The language recognition performances on the NIST LRE07 30-second are presented in Figure 7.7.

From the results, we find that the combined merit outperforms the two individual criteria across all of the selection points. The combined feature selection criterion takes both the language dependency and discriminative characteristics into consideration, hence better language recognition performance are achieved.

7.3.5 Incorporating higher order n -gram features in language recognition

With the above mentioned feature selection methods, the dimension of the trigram feature vectors derived from the LRE07 test set is reduced to less than half of its original size without affecting the language recognition performance (from 88292 to 40000). With

Figure 7.5: Mean difference of ranked features derived by the Chi-squared criterion on the LRE07 30-second set

N-gram order	LRE07 30-second		LRE03 30-second	
	EER(%)	# features	EER(%)	# features
Baseline 3-gram	2.11	88292	1.35	83165
SM 3-gram	2.10	40000	1.03	30000
SM 4-gram	1.84	279299	0.92	257312
CM 3-gram	2.08	37000	1.04	30000
CM 4-gram	1.88	264396	0.95	259871

Table 7.2: EER (%) with different order of n -gram and the number of selected features

a smaller feature dimension, incorporating more lexical information by expanding the features into higher order becomes feasible. For example, in the baseline system, incorporating up to trigram n -gram statistics leads to 88292 features. If we include the 4-gram phone statistics, the feature space will be increased to about 1273000. This feature dimension is too huge for a SVM classifier training. After the feature selection, we can incorporate only those 4-grams that are expanded from the selected features, the actual dimension of the 4-grams will be manageable.

In this section, we conduct a series of language recognition experiments by incorporating higher order n -gram features derived from the selected features. Given the high discriminative features selected with the proposed methods, we expand them into 4-gram

Figure 7.6: EER (%) with different number of features selected by the Chi-squared criterion on the LRE07 30-second set

phonotactic features. That is, if we use x_i to present any phone in the phone inventory, when a trigram feature element $x_1x_2x_3$ is selected in the feature selection step, in the expansion step, its derived 4-gram features $x_ix_1x_2x_3$ and $x_1x_2x_3x_i$ will be included in language recognition.

Table 7.2 shows the language recognition results of 3-gram feature vectors and derived 4-gram feature vectors. The numbers of selected features, which are the actual dimension of the resulting feature vectors, are also shown in the table. The feature selection is conducted with both SVM-margin and Chi-squared criteria. The 4-gram features are derived from those 3-gram features which have been selected through feature selection.

The performance of the proposed two feature selection methods are also compared in Table 7.2. Comparable performances are obtained by applying the two feature selection criteria on feature selection. The experiment results show that the language recognition performance is improved with the higher order n -gram features. In conclusion, the feature dimension is reduced using the proposed feature selection methods, and the inclusion of higher order n -gram features achieves better language recognition accuracy than original feature set.

Figure 7.7: EER (%) with different feature selection criteria

7.4 Summary

This chapter examines methods to select phonotactic feature with the criteria of separation margin and Chi-squared measure for spoken language recognition. We propose to use the minimum mean difference (MD) point to determine the decision threshold of features selection. With the proposed feature selection strategy, lower dimensional features are obtained without affecting the language recognition performance. Higher order n -gram features can then be derived from the selected features and incorporated into the language modeling process, and thus a better language recognition accuracy can be achieved.

The two feature selection criteria measure the merit of a feature in language recognition from different point of views. We take advantage of merit measure from different views by combining the two proposed feature selection criteria, the language recognition performance is further improved compared to the individual criterion.

Chapter 8

Conclusions

This thesis examined methods to enhance the phonotactic feature for high performance language recognition. Specifically, my contributions are:

- Target-oriented front-end: a target-oriented method is proposed to incorporate phone discriminative abilities in constructing target languages oriented tokenization front ends. Multiple target-oriented phone tokenizers are derived, hence richer phonotactic information is obtained for language recognition;
- Target-aware language models: a target-aware language model method is proposed to construct a new phone tokenizer for each target language by emphasizing on those phones that have higher discriminative abilities in the language model;
- Language recognition with relevance feedback: automatic query expansion techniques are applied in language recognition to augment the original test queries to improve the language recognition performance on short test segments;
- Phonotactic feature selection: a feature selection method based on two different feature selection criteria is proposed to eliminate the redundant information for efficient language recognition.

This chapter summarizes the contributions of the thesis and discusses possible future research directions.

8.1 Contributions

8.1.1 Target-oriented front-end

A novel target-oriented phone tokenizer (TOPT) construction approach is proposed. We construct a set of new phone tokenizers from existing phone recognizers without requiring new transcribed training data. For each of the target languages, the TOPT method selects a subset of phones to form a new phone tokenizer based on their discriminative ability. Two different phone selection criteria for constructing TOPTs are proposed: separation margin (SM) and mutual information (MI), they achieved comparable performance in language recognition experiments. A TOPT selection method to short-list the adequate number of TOPT candidates is proposed to improve system efficiency.

The proposed target-oriented front-end construction methods is novel in generating additional phone tokenizers without requiring additional annotated speech samples. Using the TOPT method, the phone inventory size in the derived phone tokenizer is reduced, and the use of higher order phonotactic features such as four-gram phone statistics becomes feasible. By incorporating more phone tokenizer front-ends and higher n -gram statistics, the phonotactic resolution is increased, which leads to a significant improvement in language recognition performance, as shown in the experimental results on the NIST LRE96, LRE03 and LRE07 tasks. The results also reveal that TOPTs derived from UPRs outperform those derived from language-specific phone recognizers.

8.1.2 Target-aware language models

An alternative phone tokenizer construction method is proposed: target-aware language models (TALM). It incorporates the discriminative ability of each phone in the decoding process, hence multiple phone tokenizers are derived as front-end for language recognition. Each TALM phone tokenizer has the same set of acoustic models but different language model.

Target-aware language models (TALM) can be seen as an extension of the target-oriented phone tokenizer (TOPT) construction method. The TALM method does not eliminate any phone, but constructs a new language model according to the discriminative ability of each phone. For each target language, a TALM tokenizer is constructed with

its unique target-aware language model. The experiment results show that the TALM derived tokenizer outperforms TOPT derived tokenizer when keeping the same number of phones in their phone inventory.

8.1.3 Language recognition with relevance feedback

Relevance feedback techniques are applied in language recognition in a fully automatic mode by comparing the similarity of the test query and utterances in the reference corpus. In the query expansion process, we assume the top ranked reference utterances are relevant and the bottom ranked utterances are irrelevant to the test utterance. We proposed to revise the n -gram statistics of the original test utterance with relevant and irrelevant utterances. In this way, a new test query can be obtained which has richer phonotactic information.

Three automatic speech query expansion methods are studied for spoken language recognition. The experiment results show that all three methods are effective in the short length automatic language recognition task. By utilizing a set of training data from each of the target language as reference corpus, the automatic query expansion method provides a way to improve language recognition performance at a low cost. The proposed automatic relevance feedback technique reduces the relative equal error rate by 16.25%, 4.94% and 10.15% on the NIST LRE 1996, 2003 and 2005 databases respectively for 3-second trials.

8.1.4 Phonotactic feature selection

Two different phonotactic feature selection methods with the criteria of separation margin and Chi-squared test [37] are proposed for language recognition. The two criteria rank the phonotactic features by measuring their contribution to the language separation from different point of views. A novel method: minimum mean difference (MD) point measure is proposed to select the optimal threshold for phonotactic feature selection.

With the proposed feature selection strategy, much lower dimensional features are obtained without affecting the language recognition performance. Higher order n -gram features can then be derived from the selected features and incorporated into the language modeling process, thus better language recognition performance can be achieved.

8.2 Conclusions and future works

The phonotactic and acoustic features are two most effective features in spoken language recognition. Although this thesis studies the approaches for high performance phonotactic feature based language recognition, similar techniques are applicable to acoustic feature based language recognition systems. For example the TOPT phone selection strategy may be applied to acoustic feature selection as different acoustic features may have different discriminative ability in language separation. We may also be able to build a target-oriented acoustic feature classifier by only keeping those acoustic features with higher discriminative ability in separating one language from others.

In the TALM method, the unigram and bigram statistics are proved to be useful in deriving target-aware language model. In the future works, we would like to study the way to derive TALM with higher n -gram language models. We have shown that TALM outperforms target-oriented phone tokenizers (TOPT) which doesn't employ language model during phone decoding in the PPR-VSM paradigm. The same technique is readily applicable to PPR-LM and other phonotactic language recognition systems.

The query expansion approach is shown to be effective on the short length test queries. In this thesis, the query expansion technique is utilized in the context of phonotactic vector based language recognition. Without loss of generality, the technique can be extended to any other common language recognition frameworks, such as acoustic GMM, spectral SVM, with only a minor modification of the query expansion procedure. Query expansion provides a way to improve automatic language recognition performance at a low cost. Nonetheless, the quality of the reference corpus may affect the relevance feedbacks. It is important to choose a reference corpus that is close to the test task in terms of acoustic or channel conditions. We will extend the proposed methods towards other language recognition frameworks and study the robustness issues of the query expansion in the future works.

Two phonotactic feature selection criteria have been studied in this thesis. Features selected from SVM-margin and Chi-squared methods achieve comparable performance in language recognition. The Chi-squared method measures the individual features contribution to the language separation, while the SVM-margin method considers the contribution of each feature in the group of features. Combining the two criteria, the selected

features achieved better performance in language recognition. In future works, we would like to explore ways to combine other methods to select phonotactic features for language recognition.

Appendix A

Corpus

In this appendix, we will describe the language corpus for the experiments. It includes development data for language classifiers training and test corpus. We wish to highlight that all the development and test data are pre-processed by an energy based speech activity detection program to remove silence and then chunked into smaller speech segments before they are fed into the phone recognizers for decoding.

A.1 Language corpus for training

In this thesis, all the experiments are performed using the PPR-VSM system as described in Section 3.2. Three corpuses are used for language classifier training: Linguistic Data Consortium (LDC) CallFriend [104] contains languages/dialects from 12 languages, OHSU 2005 [105] contains languages/dialects from 8 languages, and the LRE07 development data sets released by LDC contains 6 languages/dialects. Table A.1 summarizes the languages/dialects included in these databases.

In CallFriend corpus, the data for each language are grouped into three portions: train, devtest and evaltest. As some of the testing data in LRE96 are extracted from the ‘evaltest’ set, we only use ‘train’ and ‘devtest’ portion as our development data, the ‘evaltest’ part is not used. Specifically, the ‘Train’ set is used to build the SVMs in the VSM back-end and the ‘devtest’ set is used to train the GMM models for the final language classifier.

For LRE96, LRE03 and LRE05, we had only used the LDC CallFriend corpus in the training process. In the LDC CallFriend corpus, two dialects for each of the following

Data	Languages
CallFriend	English (southern, non-southern), Mandarin (Mainland, Taiwan), Korean, Japanese, Vietnamese, Hindi, French, Arabic, Farsi, German, Tamil, Spanish (Caribbean, non-Caribbean)
OHSU 2005	English (American, Indian), Hindi, Japanese, Korean, Mandarin (Mainland, Taiwan), Tamil, Spanish, German
LRE07 Dev	Arabic, Bengali, Chinese (Min, Wu, Cantonese), Russian, Thai, Urdu

Table A.1: Languages/dialects in the development data

languages are included: English, Mandarin and Spanish. In the development process, we treat the dialects of English, Mandarin and Spanish as different languages. Hence there are 15 languages/dialects in total. The CallFriend corpus consists of 60 unscripted telephone conversations for each language/dialect. Each conversation is 5-30 minutes in length. In each conversation, the two involved speakers are native speakers of that language/dialect. The contents of the conversations are general without specific topic. The speech is recorded through the domestic telephone channel and the quality of the speech signal is good (with SNR value between 20dB and 25dB). In our experiments, no noise reduction process is applied on the train and test data.

In the evaluation of LRE07, the CallFriend, LRE07 development set and OHSU corpus are used for training. The LRE07 development set consists of approximately 53 hours of conversational telephone speech segments for the following 8 languages/dialects: Arabic, Bengali, Cantonese, Min Nan Chinese, Wu Chinese, Russian, Thai and Urdu. The materials are extracted from the LDC’s CALLHOME, CALLFRIEND and Mixer collections. The average SNR value of speech in LRE07 development set is calculated to be about 20-25dB. The OHSU corpus has approximately 65 hours of telephone conversations in the following 10 languages/dialects: English (American, Indian), Hindi, Japanese, Korean, Mandarin (Mainland, Taiwan), Tamil, Spanish and German. The speech was collected by Oregon Health and Science University (OHSU) [105]. The SNR value of speech in OHSU corpus is also in 20-25dB. No noise reduction is performed on both of the LRE07 development and OHSU corpus in all experiments.

Test set	Target Languages	OOS	#30s	#10s	#3s
LRE96	English, Arabic, Farsi, French, Mandarin, German, Vietnamese, Hindi, Japanese, Spanish, Korean, Tamil	NIL	1492	1502	1503
LRE03	English, Arabic, Farsi, French, Mandarin, German, Hindi, Japanese, Spanish, Korean, Tamil, Vietnamese	Russian	1280	1280	1280
LRE05	English(American, Indian), Hindi, Japanese, Korean, Mandarin (Mainland, Taiwan), Spanish (Mexican), Tamil	German	3662	3662	3662
LRE07	Arabic, Bengali, Chinese (Cantonese, Mainland, Taiwan, Min, Wu), English (American, Indian), Hindustani (Hindi, Urdu), Spanish (Caribbean, non-Caribbean), Farsi, German, Japanese, Korean, Russian, Tamil, Thai, Vietnamese	Italian, Punjabi, Tagalog, Indonesian, French	2510	2510	2510

Table A.2: Target languages/dialects and number of trials in the NIST LREs

A.2 Language corpus for testing

In this thesis, the proposed methods are evaluated on the 1996, 2003, 2005 and 2007 NIST Language Recognition Evaluation (LRE) tasks [41]. As described in Chapter 2, the test trials in the NIST LRE are grouped into three categories according to their lengths: 30 seconds, 10 seconds and 3 seconds, while the 30 second trials are the core tasks in the NIST LRE. In these four NIST language recognition evaluations (1996-2007), the evaluations focus on language and dialect detection in the context of conversational telephone speech.

For each NIST LRE task, the target languages concerned are different. There are 12 target languages in the 1996 NIST LRE (LRE96) and 2003 NIST LRE (LRE03), 7 target languages in 2005 NIST LRE (LRE05) and 14 target languages in 2007 NIST LRE (LRE07). Table A.2 summarizes the collection of target languages and out-of-set languages (OOS) as well as the numbers of trials in the four NIST LREs. Dialects or

accents are involved in some languages, as shown in the brackets. The NIST LRE96 test set is drawn from the ‘evaltest’ portion of the LDC CallFriend corpus, and it is a closed set language recognition task in which no out-of-set language (OOS) is involved. While for the later years (LRE03, LRE05 and LRE07), the language recognition evaluations are open-set tasks and the number of OOS trials is increasing.

Each speech signal in the NIST LRE test data is a single side of a 4-wire conversation recording represented in 8-bit 8-kHz mu-law format. For LRE03, most of the conversations are drawn from the LDC CallFriend corpus. For LRE05 and LRE07, the evaluation data are drawn primarily from conversations collected by the Oregon Health & Science University (OHSU). The gender information for LRE96 is not revealed. The female to male ratios are 1.27:1, 1.36:1 and 1.71:1 for LRE03, LRE05 and LRE07 respectively. Most of the speakers are speaking in their first language/dialect.

References

- [1] E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, W. M. Campbell and D. A. Reynolds, “Acoustic, phonetic and discriminative approaches to automatic language recognition,” *Eurospeech 2003*, pp. 1345-1348, 2003.
- [2] R. Tong, B. Ma, D. Zhu, H. Li and E. S. Chng, “Integrating acoustic, prosodic and phonotactic features for spoken language identification,” *ICASSP 2006*, pp. 205-208, 2006.
- [3] V. W. Zue and J. R. Glass, “Conversational interfaces: advances and challenges,” *Proceedings of the IEEE*, Vol. 88, No. 8, pp. 1166-1180, 2000.
- [4] A. Waibel, P. Geutner, L. M. Tomokiyo, T. Schultz and M. Woszczyna, “Multilinguality in speech and spoken language systems,” *Proceedings of the IEEE*, Vol. 88, No. 8, pp. 1181-1190, 2000.
- [5] N. Bertoldi and M. Federico, “Cross-sanguage spoken document retrieval on the TREC SDR collection,” *Proc. CLEF 2002: Workshop on Cross-Language Information Retrieval and Evaluation*, Rome, 2002. Springer Verlag.
- [6] Y. K. Muthusamy, N. Jain and R. A. Cole, “Perceptual benchmarks for automatic language identification,” *ICASSP 1994*, pp. 333-336, 1994.
- [7] T. J. Hazen and V. W. Zue, “Recent improvements in an approach to segment-based automatic language identification,” *ICSLP 1994*, pp. 1883-1886, 1994.
- [8] M. A. Zissman, “Comparison of four approaches to automatic language identification of telephone speech,” *IEEE Transactions on Speech and Audio Processing*, Vol. 4, No. 1, pp. 31-44, 1996.

REFERENCES

- [9] Y. K. Muthusamy, E. Barnard, R. A. Cole, “Automatic language identification: a review/tutorial,” *IEEE Signal Processing Magazine*, October, 1994.
- [10] L. Burget, P. Matejka and J. Cernocky, “Discriminative training techniques for acoustic language identification,” *ICASSP 2006*, pp. 209-112, 2006.
- [11] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds and J. R. Deller Jr., “Approaches to language identification using Gaussian mixture models and shifted delta cepstral features,” *ICSLP 2002*, pp. 33-36, 2002.
- [12] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer and P. A. Torres-Carrasquillo, “Support vector machines for speaker and language recognition,” *Computer Speech and Language*, Vol. 20, pp. 210-229, 2006.
- [13] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, “Compensation of nuisance factors for speaker and language recognition,” *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, pp. 1969-1978, 2007.
- [14] M. Sugiyama, “Automatic language recognition using acoustic features,” *ICASSP 1991*, Vol. 2, pp. 813-816, 1991.
- [15] K. Kirchhoff, S. Parandekar, and J. Bilmes, “Mixed memory Markov models for automatic language identification,” *ICASSP 2002*, Vol. 1, pp. 761-764, 2002.
- [16] Andre G. Adami, Hynek Hermansky, “Segmentation of speech for speaker and language recognition,” *Eruospeech 2003*, pp. 841-844, Geneva, 2003.
- [17] Obuchi, Y. Sato, N., “Language identification using phonetic and prosodic HMMs with feature normalization,” *ICASSP 2005*, pp. 569- 572, March, 2005.
- [18] Mary Leena , B. Yegnanarayana, “Extraction and representation of prosodic features for language and speaker recognition,” *Speech Communication*, Vol. 50, no. 10, pp. 782-796, 2008.
- [19] B. Yin, E. Ambikairajah, F. Chen, “Voiced/unvoiced pattern-based duration modeling for language identification,” *ICASSP 2009*, pp. 4341-4344, 2009.

REFERENCES

- [20] P. Matejka, P. Schwarz, J. Cernocky, P. Chytil, “Phonotactic language identification using high quality phoneme recognition,” *Interspeech 2005*, pp. 2237-2240, 2005.
- [21] H. Li, B. Ma, and C.-H. Lee, “A vector space modeling approach to spoken language identification,” *IEEE Transaction on Audio, Speech and Language Processing*, Vol. 15, No. 1, pp. 271-284, 2007.
- [22] L. Rabiner, B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Inc. N.J., USA, 1993.
- [23] C. Corredor-Ardoy, J. L. Gauvain, M. Adda-Decker, L. Lamel, “Language identification with language-independent acoustic models,” *Eurospeech 1997*, pp. 55-58, 1997.
- [24] P. Matejka, L. Burget, P. Schwarz, and J. Cernocky, “Use of anti-models to further improve state-of-the-art PRLM language recognition system,” *ICASSP 2006*, pp. 197-200, 2006.
- [25] J. L. Gauvain, A. Messaoudi, and H. Schwenk, “Language recognition using phone lattices,” *ICSLP 2004*, pp. 1283-1286, 2004.
- [26] B. Ma and H. Li, “Spoken language identification using bag-of-sounds,” *International Conference on Chinese Computing (ICCC 2005)*, pp. 21-27, 2005.
- [27] K. M. Berkling, and E. Barnard, “Analysis of phoneme-based features for language identification,” *ICASSP 1994*, pp. 289-292, 1994.
- [28] Y. Yan, and E. Barnard, “An approach to automatic language identification based on language dependent phone recognition,” *ICASSP 1995*, Vol. 5, pp. 3511-3514, 1995.
- [29] K. M. Berkling, and E. Barnard, “Language identification of six languages based on a common set of broad phonemes,” *ICSLP 1994*, pp. 1891-1894, 1994.
- [30] K. M. Berkling, T. Arai and E. Barnard, “Analysis of phoneme-based features for language identification,” *ICASSP 1994*, pp. 289-292, 1994, Japan.

REFERENCES

- [31] Kumar, C. P. Santhosh, H. Li, R. Tong, P. Matejka, L. Burget, J. Cernocky, "Tuning phone decoders for language identification," *ICASSP 2010*, pp. 5010-5013, March, 2010, USA.
- [32] P. Torres-Carrasquillo, E. Singer, W. Campbell, T. Gleason, A. McCree, D. Reynolds, F. Richardson, W. Shen and D. Sturim, "The MITLL NIST LRE 2007 language recognition system," *Interspeech 2008*, pp. 718-722, 2008, Australia.
- [33] R. Tong, B. Ma, H. Li and E. S. Chng, "A target-oriented phonotactic front-end for spoken language recognition," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 17, issue 7, pp. 1335-1347, Sept, 2009.
- [34] R. Tong, B. Ma, H. Li and E. S. Chng, "Target-aware language models for spoken language recognition," *Interspeech 2009*, pp. 200-203, 6 - 12 September 2009, UK.
- [35] R. Tong, H. Li and B. Ma, "Spoken language recognition with relevance feedback," *ICASSP 2007*, pp. 861-864, 2007, USA.
- [36] F. S. Richardson and W. M. Campbell, "Language recognition with discriminative keyword selection," *ICASSP 2008*, pp. 4145-4148, 2008, USA.
- [37] R. Tong, B. Ma, H. Li and E. S. Chng, "Selecting phonotactic features for language recognition," *Interspeech 2010*, pp. 737-740, September, 2010, Japan.
- [38] A. S. House and E. P. Neuberg, "Toward automatic identification of the language of an utterance. I. Preliminary methodological considerations," *Journal of the Acoustical Society of America*, 62(3), pp. 708-713, 1977.
- [39] M. A. Zissman, "Automatic language identification using Gaussian mixture and hidden markov models," *ICASSP 1993*, pp. 399-402, Minneapolis, MN, April 1993.
- [40] Alvin Martin, George Doddington, Terri Kamm, Mark Ordowski, Mark Przybocki, "The DET curve in assessment of detection task performance," In *Eurospeech 1997*, pp. 1895-1898, 1997, Rhodes, Greece.
- [41] <http://www.itl.nist.gov/iad/mig/tests/lre/>.

REFERENCES

- [42] K. P. Li and T. J. Edwards, “Statistical models for automatic language identification,” *ICASSP 80*, pp. 884-887, Denver, April, 1980.
- [43] Yeshwant K. Muthusamy, Ronald A. Cole, and Murali Gopalakrishnan, “A segment-based approach to automatic language identification,” *ICASSP 1991*, pp. 353-356, 1991.
- [44] Lori F. Lamel and Jean-Luc Gauvain, “Cross-lingual experiments with phone recognition”, *ICASSP 1993*, pp. 507-510, 1993.
- [45] L. F. Lamel and J. L. Gauvain, “Language identification using phone based acoustic likelihoods”, *ICASSP 1994*, pp. 293-296, 1994.
- [46] T. Schultz , I. Rogina , A. Waibel, “Experiments with LVCSR based language identification,” *Speech Research Symposium (SRS) 1995*, pp. 89-92, 1995.
- [47] M. A. Zissman and E. Singer, “Automatic language identification of telephone speech messages using phoneme recognition and n-gram modeling,” *ICASSP 1994*, pp. 305-308, 1994.
- [48] J. Navratil and W. Zuhlke, “Double bigram-decoding in phonotactic language identification,” *ICASSP 1997*, Munich, Germany, April 1997, Vol. 1, pp. 1115-1118.
- [49] P. A. Torres-Carrasquillo, D. A. Reynolds, J. R. Deller, Jr, “Language identification using Gaussian mixture model tokenization,” *ICASSP 2002*, pp. 757-760, 2002.
- [50] Shen, W., Reynolds, D. A., “Improving phonotactic language recognition with acoustic adaptation,” *Interspeech 2007*, August 27-31, 2007, Antwerp, Belgium.
- [51] S. M. Siniscalchi, J. Reed, T. Svendsen, C. H. Lee, “Exploring universal attribute characterization of spoken languages for spoken language recognition,” *Interspeech 2009*, pp. 168-171, September, 2009.
- [52] M. Penagarikano, A. Varona, L. J. Rodriguez-Fuentes and G. Bordel, “Using cross-decoder phone cooccurrences in phonotactic language recognition,” *ICASSP 2010*, pp. 5034-5037, Dallas, USA, 2010.

REFERENCES

- [53] R. G. Leonard and G. R. Doddington, "Automatic language identification," *Technical Report RADC-TR-74-200*, Air Force Rome Air Development Center, August 1974.
- [54] Cimarusti, D. Ives, R., "Development of an automatic identification system of spoken languages: phase 1," *ICASSP 1982*, pp. 1661-1664, May 1982.
- [55] L. Riek, W. Mistretta and D. Morgan, "Experiments in language identification," *Lockheed Sanders, Inc., Nashua, NH, Tech. Rep. SPCOT-91-002*, 1991.
- [56] P. Kenny and P. Dumouchel, "Experiments in speaker verification using factor analysis likelihood ratios," in Proc. *Odyssey-04, The Speaker and Language Recognition Workshop*, pp. 219-226, 2004.
- [57] Alex Solomonoff, Carl Quillen, and William M. Campbell, "Channel compensation for SVM speaker recognition," *Odyssey-04, The Speaker and Language Recognition Workshop*, pp. 57-62, 2004.
- [58] F. Castaldo, S. Cumani, P. Laface, D. Colibro, "Language recognition using language factors," *Interspeech 2009*, pp. 176-180, 2009.
- [59] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process Letter*, Vol. 13, pp. 308-311, 2006.
- [60] B. Burget, P. Matejka, and J. Cernocky, "Discriminative training techniques for acoustic language identification," *ICASSP 2006*, pp. 209-212, Toulouse, France.
- [61] D. Zhu, B. Ma and H. Li, "Large margin estimation of Gaussian mixture model parameters with extended Baum-Welch for spoken language recognition," *Interspeech 2009*, pp. 2179-2182, 2009.
- [62] J. T. Foil, "Language identification using noisy speech," *ICASSP 1986*, pp. 861-864, Tokyo, Japan.
- [63] Savic, M. Acosta, E. Gupta, S. K., "An automatic language identification system," *ICASSP 1991*, pp. 817-820, April 14-17, Toronto, Canada.

REFERENCES

- [64] Y. K. Muthusamy, “A segmental approach to automatic language identification,” *Ph.D thesis, Oregon Graduate Institute of Science & Technology*, July 1993.
- [65] A. E. Thym-Gobbel and S. E. Hutchins, “On using prosodic cues in automatic language identification,” *ICSLP 96*, pp. 1768-1771, Philadelphia, USA, October 1996.
- [66] Ramus, F., Mehler, J., “Language identification with suprasegmental cues: A study based on speech synthesis,” *Acoustical Society of America Journal*, 105(1), pp. 512-521, 1999.
- [67] Fred Cummins, Felix Gers, Jurgen Schmidhuber, “Language identification from prosody without explicit features,” *Eurospeech 1999*, pp. 371-374, Budapest, Hungary.
- [68] B. Ma, D. Zhu and R.Tong, “Chinese dialect Identification using tone features based on pitch flux,” *ICASSP 2006*, 2006.
- [69] G. Salton, “The SMART retrieval system - experiments in automatic document processing,” Prentice-Hall, Inc., Upper saddle River, N.J., 1971.
- [70] M. P. Harper and M. Maxwell, “Spoken language characterization,” *Springer Handbook of Speech Processing*, J. Benesty, Y. Huang, M. M. Sondhi. Springer, pp. 797-809, 2007.
- [71] Pavel Matejka, Lukas Burget, Ondrej Glembek, Petr Schwarz, Valiantsina Hubeika, Michal Fapso, Tomas Mikolov, Oldrich Plchot, and Jan Cernocky, “BUT language recognition system for NIST 2007 evaluations,” *Interspeech 08*, Brisbane, Australia. 2008.
- [72] H. Hermansky and S. Sharma, “Temporal patterns (TRAPs) in ASR of noisy speech,” *ICASSP1999*, Phoenix, Arizona, March 1999, pp. 2427-2431.
- [73] W. M. Campbell, F. Richardson, and D. A. Reynolds, “Language recognition with word lattices and support vector machines,” *ICASSP 2007*, pp. 15-20, Honolulu, Hawaii, US, 2007.

REFERENCES

- [74] S. Nakagawa, Y. Ueda and T. Seino, "Speaker-independent, text-independent language identification by HMM," *ICSLP 92*, Vol. 2, Oct 1992, pp. 1011-1014.
- [75] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, 24(5), pp. 513-523, 1988.
- [76] <http://htk.eng.cam.ac.uk/docs/cuhtk.shtml>.
- [77] J. Navratil, "Spoken language recognition - A step toward multilinguality in speech processing," *IEEE Transactions on Speech and Audio Processing*, Vol. 9, issue 6, pp. 678-685, 2001.
- [78] K. C. Sim, H. Li, "On acoustic diversification front-end for spoken language identification," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 16, No. 5, pp. 1029-1037, 2008.
- [79] B. Ma, H. Li, and R. Tong, "Spoken language recognition using ensemble classifiers," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, issue 7, pp. 2053-2062, 2007.
- [80] W. M. Campbell, T. P. Gleason, J. Navratil, D. Reynolds, W. Shen, E. Singer, and P. A. Torres-Carrasquillo, "Advanced language recognition using cepstra and phonotactics: MITLL system performance on the NIST 2005 language recognition evaluation," *IEEE Odyssey: The Speaker and Language Recognition Workshop*, pp. 1-8, 2006.
- [81] P. A. Torres-Carrasquillo, E. Singer, W. M. Campbell, T. Gleason, A. McCree, D. A. Reynolds, F. Richardson, W. Shen, and D. Sturim, "The MITLL NIST LRE 2007 language recognition system," *Interspeech 2008*, pp. 719-722, 2008.
- [82] W. M. Campbell, J. R. Campbell, D. A. Reynolds, D. A. Jones, T. R. Leek, "High-level speaker verification with support vector machines," *ICASSP 2004*, pp. 73-76, 2004.
- [83] H. Li, B. Ma, K.-A. Lee, K.-C. Sim, H. Sun, R. Tong, D. Zhu, and C. You, "NIST 2007 language recognition evaluation: from the perspective of IIR," *22nd Pacific Asia Conference on Language, Information and Computation (PACLIC)*, 2008.

REFERENCES

- [84] R. Tong, B. Ma, H. Li and E. S. Chng, “Target-oriented phone tokenizers for spoken language recognition,” *ICASSP 2008*, pp. 4221-4224, 2008.
- [85] R. Tong, B. Ma, H. Li and E. S. Chng, “Target-oriented phone selection from universal phone set for spoken language recognition,” *Interspeech 2008*, pp. 715-718, 2008.
- [86] B. Ma, R. Tong and H. Li, “Discriminative vector for spoken language recognition,” *ICASSP 2007*, pp. 1001-1004, 2007.
- [87] G. Forman, “An extensive empirical study of feature selection metrics for text classification,” *Journal of Machine Learning Research*, 2003, pp. 1289-1305, 2003.
- [88] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, Wiley, New York, ISBN 0-471-05669-3. 1973.
- [89] Y. Yang and J. Pedersen, “A comparative study on feature selection in text categorization,” *The Fourteenth International Conference on Machine Learning*, pp. 412-420, 1997.
- [90] T. Mikolov, O. Plchot, O. Glembek, P. Matejka, L. Burget and J. Cernocky, “PCA-based feature extraction for phonotactic language recognition,” *Proc Odyssey 2010*, pp. 251-255, 2010.
- [91] M. Agarwal, N. Anerousis, M. Gupta, V. Mann, L. Mummert and N. Sachindran, “Problem determination in enterprise middleware systems using change point correlation of time series data,” *Proc of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*, Vancouver, Canada, April 2006.
- [92] T. Joachims, “Making large-scale SVM learning practical,” *Advances in Kernel Methods - Support Vector Learning*, B. Scholkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.
- [93] R. Rosenfeld, “Adaptive statistical language modeling: a maximum entropy approach,” *Ph.D. Thesis, Carnegie Mellon University*, 1992.

REFERENCES

- [94] A. Cole, Y. Muthusamy, B. Oshikal, "The OGI multi-language telephone speech corpus," *ICSLP 1992*, pp. 895-898, 1992.
- [95] T. Lander and R. Cole and B. Oshika and M. Noel, "The OGI 22 language telephone speech corpus," *Eurospeech 1995*, pp. 895-898, 1995.
- [96] J. G. Wilpon, L. R. Rabiner, and T. Martin, "An improved word-detection algorithm for telephone-quality speech incorporating both syntactic and semantic constraints," *AT&T Bell Labs. Tech. J.*, Vol. 63, pp. 479-498, Mar. 1984
- [97] H.-C. Wang, "MAT - A project to collect mandarin speech data through telephone networks in Taiwan," *Computational Linguistics and Chinese Language Processing*, Vol. 2, No. 1, pp. 73-89, 1997.
- [98] A. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, pp. 245-271, 1997.
- [99] E. N. Eftthimiadis, "A user-centered evaluation of ranking algorithms for interactive query expansion," *Proc. 16th ACM SIGIR* pp. 146-159, 1993.
- [100] S. E. Robertson, "On term selection for query expansion," *Journal of Documentation*, Vol 46, no. 4, pp. 359-364.
- [101] J. J. Rocchio, "Relevance feedback in information retrieval," *The SMART Retrieval System - Experiments in Automatic Document Processing*, G. Salton (Ed.), pp. 313-323, Englewood Cliffs, N.J. Prentice-Hall, Inc., 1971.
- [102] W. B. Croft, "A model of cluster searching based on classification," *Information Systems*, Vol. 5, pp. 189-195, 1980.
- [103] L. L. Chase, "Error-responsive feedback mechanisms for speech recognizers," *Tech. Report CMU-RI-TR-97-18*, 1997.
- [104] Linguistic Data Consortium (LDC), "The CallFriend corpra," <http://www ldc.upenn.edu/Catalog/byType.jsp>
- [105] OHSU Corpora, <http://www.ohsu.edu/>

REFERENCES

- [106] Language Identification Corpus of the Institute for Infocomm Research
- [107] R. E. Walpole and R. H. Myers, *Probability and Statistic for Engineers and Scientists*, 4th ed. New York: Macmillan, 1989.