Spoken Language Identification with Prosodic Features

NG, Wai Man

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Electronic Engineering

The Chinese University of Hong Kong
March 2011

Abstract

This thesis focuses on the use of prosodic features for automatic spoken language identification (LID). LID is the problem of automatically determining the language of spoken utterances. After three decades of research, the state-of-the-art LID systems seem to give a saturating performance. To meet the tight requirements on accuracy, prosody is proposed as alternative features to provide complementary information to LID.

There are no conventional ways to model prosody. We use a large prosodic feature set which covers fundamental frequency (F0), duration and intensity. It also considers various extraction and normalization methods of each type of features. In terms of modeling, the vector space modeling approach is adopted. We introduce a framework called prosodic attribute model (PAM) to model the acoustic correlates of prosodic events in a flexible manner. Feature selection and preliminary LID tests are carried out to derive a preferred term-document matrix construction for modeling.

The PAM-based prosodic LID system is compared with other prosodic LID systems with a task of pairwise language identification. The advantages of comprehensive modeling of prosodic features is clearly demonstrated. Analysis reveals the confusion patterns among target languages, as well as the feature-language relationship. The PAM-based prosodic LID system is combined with a state-of-the-art phonotactic system by score-level fusion. Complementary effects are demonstrated between the two different features in the LID problem. An additional operation on score calibration, which further improves the LID system performance, is also introduced.

摘要

本文主要研究韻律特徵在語言識別（LID）技術上之運用。語言識別旨在自動判別說話語句的所屬語言。歷經三十多年的研發，現時最前沿的語言識別系統似乎面對著性能飽和的問題。為迎合對系統高準確度的逼切要求，我們提出使用非傳統的韻律特徵為語言識別系統提供互補資訊。

現存並無常規方法對韻律特徵建立模型。我們將使用一大型的韻律特徵集，它涵蓋基頻、時間長度及音量強度，並考慮各種特徵的提取及正規化方法。我們以向量空間模型法進行訓練，並描述了一個韻律屬性模型（PAM），該模型可在語言識別過程中，以靈活的方式為各種韻律現象的聲學相關建模。我們進行了特徵選取以及初階的語言識別測試，以推估較佳的詞彙-文件矩陣構造。

基於韻律屬性模型（PAM）的語言識別系統將會與其他亦以韻律特徵為主的同類系統進行比對。以二元語言進行評估，結果清楚顯示全面訓練模型在語言識別上的優勢。我們透過分析，揭示了不同目標語言之間的混淆模式，以及特徵與語言之間的特定關係。基於韻律屬性模型（PAM）的語言識別系統亦會以評分融合的方法，與現時最尖端、基於音位配列結構特徵（phonotactic）的語言識別系統相結合。這兩種不同特徵在語言識別問題中顯示了很好的互補效應。本文亦另介紹了評分校準技術，其可進一步提升語言識別系統之性能。

Acknowledgements

First, I would like to express my gratitude to Prof. Tan Lee, my thesis supervisor, who has been guiding me from the time when I knew nothing about research. From him I learn impartiality in research. I am grateful for the various learning opportunities he granted me, as well as his confidence on my work. I would like to thank Dr. Cheung Chi Leung, Dr. Bin Ma and Prof. Haizhou Li from the Institute for Infocomm Research in Singapore, for sharing their expertise and their guidance in language identification. My gratitude also goes to Prof. William S.-Y. Wang. He shows me how exciting and diversified speech and language researches can be. I am deeply affected his perseverance in language research.

The work reported in this thesis cannot be completed without the help of many others. I received invaluable advice on prosody modeling from Prof. H. Fujisaki, Prof. J. Hirschberg, Prof. Helen H. Meng, Prof. G.A. Levow, Prof. Chiu-yu Tseng and Dr. Q. Yao. Prof. C.-H. Lee, Dr. Frank K. Soong inspired me the overall system design. Prof. W.-K. Ma taught me convex optimization, which is closely related to SVM and calibration in this thesis. Dr. S.W. Lee, Dr. S.K. Tang and Dr. C.-C. Leung shared with me a lot of hands-on research experience. Yujia assisted me in drafting the Chinese scripts in the thesis. Feng, David and Gary cooperated with me to nurture a group of "computing penguins" so I can run large-scale experiments. Arthur provided me with general technical supports.

Many thanks also go to my colleagues in the Digital Signal Processing and Speech Technology Laboratory. I feel good to work with fellow labmates who share the same passion for research. I have to mention my respectable language teachers and mentors, Mr. Francis Kong, Father Naylor, Mr. P. Ho, Mr. A. Deswani, Tanaka Sensei, Yuko San and Herr Wannagat, who arouse my interests in speech and languages.

Finally, I want to dedicate this thesis to my parents, my falcon granny and caque. They work much harder than I am for this thesis and almost live through the live of a researcher. Their unconditional tolerance and support to me is astounding.

Contents

1 Introduction **1**

1.1 Objectives of research 1

1.2 Thesis outline 3

2 Language identification overview **4**

2.1 Feature extraction 5

2.1.1 Short-time acoustic features 5

2.1.2 Feature normalization and adaptation 7

2.2 Acoustic approach 7

2.2.1 Gaussian mixture modeling classification 8

2.2.2 Improved methods for acoustic approach 9

2.3 Phonotactic approach 10

2.4 Prosodic features for LID 12

2.4.1 Syllable: The basic unit for prosody 12

2.4.2 Prosodic features 14

2.5 Language classification with SVM 18

3 Acoustic tokenization **23**

3.1 Syllabification algorithm 24

3.1.1 Band pass filtering 25

3.1.2 Full wave rectification and low pass filtering 26

3.1.3 Intensity peak picking 27

3.2 Experimental results in syllabification 29

3.3 Extraction of prosodic attributes 31

3.3.1 Frame estimates and segment contours 33
3.3.2 F0, intensity and duration measurements 35
3.3.3 Normalization 37
3.3.4 Regression and residue features 39
3.3.5 Summary of prosodic attributes 40

4 Statistical language modeling 42

4.1 Attribute quantization 43
4.2 Generation of prosodic tokens 44
4.3 Vector space modeling of prosodic token n-grams 45
4.3.1 Count vectors for different prosodic token n-grams 47
4.3.2 Super term-document matrix 50
4.4 Prosodic attribute model 50
4.4.1 Attribute-wise modeling in PAM 52
4.4.2 Expanding single-attribute prosodic tokens 54
4.4.3 Combining single-attribute prosodic tokens 55
4.4.4 A preliminary design of term-document matrix under PAM 57
4.5 Attribute selection in term-document matrices 57
4.5.1 Mutual information evaluation 58
4.5.2 Language-independent prosodic attribute selection 61
4.5.3 The 14-attribute prosodic feature set 64

5 Language identification experiments 65

5.1 Detection scores from language classifier 66
5.1.1 Generation of detection scores 66
5.1.2 Backend score processing 70
5.2 Performance evaluation 71
5.3 Pairwise language identification 72
5.4 Language detection 73
5.4.1 About NIST LRE 2009 74
5.4.2 Comparison among different attribute groups 77
5.4.3 The use of skipping n-grams 78

5.4.4 The use of prosodic token boundary n-grams 79

5.4.5 Long-range prosodic n-grams 80

6 Analysis on prosodic language identification 81

6.1 LID result analysis 81

6.1.1 Performance evaluations on different target languages . . 82

6.2 Language-specific prosodic properties 87

6.2.1 Bin-level mutual information evaluation 87

6.2.2 Interpretation of the metrics 88

6.2.3 Normalization of evaluation metrics 89

6.2.4 Methodology 90

6.2.5 Language-specific prosodic properties 90

7 Score fusion and calibration 92

7.1 Application-independent score fusion 92

7.1.1 Score fusion with a phonotactic LID system 93

7.2 Target dependent score calibration 95

7.2.1 Evaluation metrics 96

7.2.2 Problem formulation 98

7.2.3 Experimental results 101

8 Conclusions 108

8.1 Summary 108

8.2 Contribution of the work 109

8.3 Limitations and Future work 110

Bibliography 112

Glossary 124

List of Figures

2.1 *Extraction of acoustic features, adopted from [1]* 6

2.2 *PPRLM block diagram adopted from [2] with target languages being Farsi, French and Tamil* 11

2.3 *F0 and intensity contours and portions within a pseudosyllable* . 16

2.4 *The binary-class training data, hyperplane and the margin* . . . 19

3.1 *A phonotactic LID approach for prosodic features* 23

3.2 *Detecting the syllable nucleus of* $/ts^hy:/$ 24

3.3 *Full wave rectification of the vocalic component of speech for nuclei detection. (left) Carrier of vocalic content modulated by the intensity profile (right) Full wave rectified signal for the demodulation of the intensity profile* 26

3.4 *Using a moving window, the peak picking algorithm rejects some local maxima in the intensity profile in syllabification* 27

3.5 *Prosodic attribute extraction* 32

3.6 *Derivation of F0 and intensity segment contours under PPS* . . 35

3.7 *Basic attributes in F0, duration and intensity types* 36

3.8 *Regression and residue attributes in F0 and intensity types* . . . 39

4.1 *Statistical language modeling of the tokenized prosodic features* . 43

4.2 *Term-document matrices for utterances spoken in two languages* n_1 *and* n_2 46

4.3 *Concatenation to form a super term-document matrix, subscripts to i are attribute indices in Table 3.2* 51

5.1 *System diagram of the prosody-based language recognizer* 66

6.1 *A directed graph for tone language pairs with high confusion. In the detection of a target language [head of the arc], a high rate of false alarm is found with testing samples of the confusing language [tail of the arc]. Arcs are all directed from tails to heads.* . . 85

6.2 *A directed graph for non-tone language pairs with high confusion.* 86

6.3 *Mutual information metrics reflected in an entropy equation* . . 89

7.1 *Likelihood ratio $\lambda^{n_t}_{\neg n_t}$ for n_t detection in a data set with two classes: n_t, n_n* 97

7.2 *System diagram of score calibration* 101

7.3 *Original and calibrated C_{min}, $C_{p\text{-}eer}$ with optimal α_{n_t,n_r} under different values of υ* 102

7.4 *DET plot for the original and the calibrated scores with $\upsilon = 3.5$* 103

List of Tables

2.1 *Comparison of the prosodic features used for different tasks* . . . 15

3.1 *Experimental results on syllabification conditioned on the width of moving window and the weight factor* 30

3.2 *The unified notation for all prosodic attributes* 41

4.1 *The defining attributes of a prosodic phone* 53

4.2 *EER with different models for NIST LRE 2009* 54

4.3 *EER with increased resolution vs trigram modeling (NIST LRE 2009)* 55

4.4 *EER with different combinations of prosodic tokens (NIST LRE 2009)* 57

4.5 $\overline{z_{(n_t,i)}}$ *scores for different normalization and extraction methods* . 62

4.6 *Selected attribute pairs to combine as prosodic tokens* 64

5.1 *Pairwise LID accuracy with OGI-TS 45-second speech* 73

5.2 *Target languages and channel conditions to be covered in LID model training* 74

5.3 *Selected prosodic attributes in feature sets of different size* 76

5.4 *EER with different attribute groups excluded (NIST LRE 2009)* 78

5.5 *EER with full vs skipping trigrams (NIST LRE 2009)* 79

5.6 *EER with different boundary n-grams (NIST LRE 2009)* 80

6.1 *Target languages/dialects of NIST LRE 2007* 82

6.2 *EER of 21 detectors for NIST LRE 2007* 83

6.3 *Language pairs with high confusion rate (>40%) for NIST LRE 2007* 84

7.1 *EER when different prosodic feature sets are fused with PPRVSM in the score level (NIST LRE 2009)* 94

7.2 *Error statistics before and after score calibration* 104

7.3 *Pooled EER for different targets* 105

7.4 *Average EER and pooled EER before and after target dependent calibration (NIST LRE 2009)* 107

List of Symbols

b	(In the context of support vector machine) Bias term which translates the position of the separating hyperplane
$b_k(\boldsymbol{v}_t)$	(In the context of acoustic LID) Multi-variate Gaussian densities
$c(k)$	The language label of trial (speech segments) k
$\boldsymbol{f}$	$[f[0], f[1], ..., f[T-1]]^T$, segment contour with T frame estimates
H	Separating hyperplane in the support vector machine
i	Index of prosodic attributes, $i \in [1 \dots I]$
K	Total number of trials (speech segments) in training/test set
k	Index to an LID training/test trial (speech segment)
$k \in \mathcal{I}(n_t)$	Set of speech segment indices in language n_t
$k \in \tilde{\mathcal{I}}(n_t)$	Estimated set of speech segment indices in language n_t
$k \in \mathcal{M}(n_t)$	Subset of $\mathcal{I}(n_t)$ with false rejection from n_t (Detection misses)
$k \in \mathcal{F}(n_t, n_r)$	Subset of $\mathcal{I}(n_r)$ with false acceptance to n_t (False alarms)
k	(In the context of acoustic LID) Mixture index in a Gaussian mixture model
L_{n_t}	A binary random variable indicating whether the speech segment belongs to n_t
M	The highest order of polynomial regression in the construction of a regression feature
N	Total number of target languages in an LID task
n	(In the context of syllabification): Number of consonants in C$^{\mathrm{n}}$V
n	(In the context of n-gram modeling): Order of n-gram
n	(In the context of attribute/term selection, acoustic LID and language recognition): Index to a target language

n_t	Target language in a specific language detector
n_r	A related language with respect to a specific language detector
P	Number of terms in term-document matrix/ Dimension of feature vector in support vector machine training
$Q_\imath$	A random variable mapping to $q_{(\imath,s)}$ with sample space $\mathcal{Q}_\imath$ (i.e. $q_{(\imath,s)} \in \mathcal{Q}_\imath$)
$q_{(\imath,s)}$	Scalar quantized value of $v_{(\imath,s)}$
s	Index to a pseudosyllable
t	Index to a frame
$\boldsymbol{v}_t$	(In the context of acoustic LID) Feature vector at frame index t
$v_{(\imath,s)}$	Feature value of prosodic attributes indexed i obtained from the segment contour of pseudosyllable s
$\boldsymbol{w}$	(In the context of support vector machine) The normal vector of the separating hyperplane
w_k	(In the context of acoustic LID) Weight of the k^{th} Gaussian mixture
w_s	Prosodic token at pseudosyllable indexed s
$\boldsymbol{x}$	Count vector of all prosodic token n-grams
α_{n_t,n_r}	Combination weight in detection target dependent calibration for target language n_t and related language n_r
Θ	The set of Gaussian mixture model parameters in acoustic LID
θ	Target independent detection threshold
θ_n	Detection threshold in the detector to language n
$\kappa^{n_t}_{\neg n_t}$	Log-likelihood ratio to language n_t from the single language detector
$\lambda^{n_t}_{\neg n_t}(k)$	Log-likelihood ratio for trial k from the detector to language n_t, combining the results from multiple language detectors
$\lambda'^{n_t}_{\neg n_t}$	Adjusted log-likelihood ratio by score calibration
#	Short pause

Chapter 1

Introduction

1.1 Objectives of research

Automatic spoken language identification (LID) is the process of automatically determining the language of spoken utterances [3, 4]. It has many applications in multi-lingual multi-media information processing. For example, an interactive voice response (IVR) system catering customers speaking different languages can use a frontend language identification module to route incoming calls [3]. With the internet and the trend of globalization, massive amount of multi-media contents in different languages become available. Voice-based user interface and dialog systems with language identification capabilities become technically feasible. This application is also attractive because it does not assume a known language [5]. Some governments are also interested in spoken language identification due to its potential in monitoring and surveillance [6].

The study of spoken language identification can be traced back to the sixties in the last century [7, 8]. The main objective of these studies is to look for language dependent characteristics in phones, intonation or other speech units. Conventional LID approaches make use of the acoustic and/or the phonotactic properties in speech. After three decades of research, these approaches attain very competitive performance in large-scale LID tasks [9]. In a typical language detection task with 30-second test utterances, the equal error rates is normally below 3%.

In recent years, conventional LID approaches seem to give a ceiling performance. Apart from striving for even higher accuracies, it would be interesting if we could find alternative approaches to the conventional LID strategies. Prosody refers to the rhythmic and intonational properties of speech [10]. It is mainly realized by the isochronous recurrence of some types of speech units, by the fundamental frequency (F0), or by intensity. A number of studies showed both human and computers are capable of distinguishing languages with prosodic cues [7, 11, 12]. Being very different from the acoustic-phonetic features, prosody features appear to be good candidates of alternative LID features. Nevertheless, there exists no standard feature set on a par with the cepstral features for automatic speech recognition. Many different prosodic features were reported in previous studies. Particularly for LID, the use of prosodic features has not been studied in a systematic way. As a result, prosodic features were generally not considered to be effective for LID.

In this thesis, we will look into the use of prosodic features for automatic spoken language identification in a systematic way. By reviewing the different prosodic features proposed for various purposes, we investigate the modeling of a comprehensive set of prosodic features for LID. These attributes are derived to represent the F0 and intensity contours, and the segmental durations in many different ways.

Under the principle of comprehensive modeling, a large number of candidate attributes are obtained for LID application. We propose to analyze the candidate attribute using an information-theoretic approach. The analysis serves two purposes. First, it facilitates a feature selection process by comparing the language-discrimination abilities of attributes. In this way, the problem dimension can be flexibly controlled. Second, the analysis can greatly improve our understanding about the prosodic characteristics of specific languages.

To draw a convincing result, the proposed prosodic features will be tested with large-scale standard LID tasks such as NIST LRE 2007 and NIST LRE 2009 [13, 14]. We will also look at the error reduction a prosodic LID system brings to a state-of-the-art phonotactic LID system. If error reduction is significant, it proves complementary effects exist between the prosodic and the conventional LID approaches.

1.2 Thesis outline

In Chapter 2, an **overview** to LID approaches would be given. It would cover the conventional acoustic and phonotactic approaches, as well as various issues in a prosody-based LID system. Chapter 3 and 4 discuss two important LID system components, namely **acoustic tokenization** and **statistical language modeling**. Chapter 5 summarizes the system flow, the scoring mechanisms and the **performance of the prosody-based LID system**. It is followed by some language- and feature-specific **analysis** in Chapter 6. Chapter 7 introduces **score fusion** with a state-of-the-art phonotactic LID system, as well as **score calibration** towards competitive LID results. The whole thesis is concluded in Chapter 8.

Chapter 2

Language identification overview

The study of spoken language identification (LID) can be traced back to the sixties in the last century [7, 8]. The main objective of these studies is to look for language dependent characteristics in phones, intonation or other speech units. Starting from the seventies, automatic LID systems were used to compare the spectral similarity between a test speech, whose language identity is in question, and some exemplars in a known language [15].

A spoken language can be identified using information gathered from different sources. Conventional LID approaches roughly fall into two large categories - acoustic and phonotactic. LID systems with the acoustic approach (also known as spectral systems) find the language-relevant properties in static features, which are derived from acoustic frames, raw waveform features, formant vectors, etc. LID systems with the phonotactic approach (also known as token systems) are based upon phone tokenization results. The co-occurrence statistics of sequences of allowable phones and phonemes in different languages are modeled to achieve LID. There is a rich source of literature introducing the LID implementations with acoustic and/or phonotactic approaches [2, 3, 5, 15, 16].

Prosody is an important component of human speech. It refers to the rhythmic and intonational characteristics, which are observed over a relatively long time span [10]. The use of prosodic features for LID has been studied sporadically over the years. Given the inferior performance compared with the conventional acoustic/phonotactic LID systems, there is no consensus about the general effectiveness of prosodic features to LID. Nevertheless, speech prosody is highly language-dependent and expected to play a role in distinguishing languages.

The first half of this chapter introduces the conventional LID approaches. Section 2.1 introduces **feature extraction**, followed by the descriptions of the **acoustic** and **phonotactic** LID approaches in Section 2.2 and 2.3 respectively. In the second half of this chapter, we will look at the use of **prosodic** features in LID in Section 2.4. Principles for using prosodic features, syllabification, and a review on different features previously used will be included. To go through the LID system completely, we will look at the **language classifier** in Section 2.5. Focus will be put on **support vector machine** (SVM), which is widely adopted in the acoustic, phonotactic approaches of LID. SVM will also be implemented in the prosody-based LID system proposed in this thesis.

2.1 Feature extraction

Modeling in both acoustic and phonotactic LID systems starts with *short-time acoustic features*. In this section the details of acoustic feature extraction is introduced. Various techniques in feature normalization, compensation and adaptation will also be highlighted.

2.1.1 Short-time acoustic features

There are several short-time acoustic features typically used for LID. Due to its popularity, Mel-scale frequency cepstral coefficients (MFCC) would be taken as an example to illustrate feature extraction below. First, short-time temporal frames of 20ms wide are obtained. 10ms shift is present between successive frames. Stationarity is assumed in all frames, and to each of which discrete fourier transform (DFT) and Mel-scale filterbank are applied. DFT transforms the acoustic signal into frequency domain representation. Mel-scale filterbank are specially designed to stimulate the non-uniform frequency resolution in the perceptual processing performed by human ears. Instead of the Mel-scale filterbank, sometimes gammatone filterbank [17] is used as it is reported to give a good approximation of the human auditory filter.

MFCC are obtained by taking the inverse DFT on the logarithm of the magnitude of the filterbank outputs. These coefficients can be seen as information about rate of change in magnitude across different spectrum bands. Formant frequencies in speech, which directly reflect the phoneme being spoken, are modeled. An overview of speech signal analysis is found in [1]. Figure 2.1 summarizes the acoustic feature extraction procedures.

Voice activity detection (VAD) is an optional preprocessing step in the LID system. It removes silence frames, which deteriorate the effectiveness of HMM modeling, and retains only the high quality speech frames for LID. In NIST Language Recognition Evaluation (LRE) 2009, radio broadcast speech was introduced in the VOA training/testing corpus. In such case, a more sophisticated VAD helped to remove also the music from the audio archive. VAD can be implemented with energy features [18], or with a phone recognizer [19].

Besides MFCC, another popular coding method in LID systems is to use Perceptual Linear Prediction (PLP) coefficients [20]. This method uses all-pole spectral modeling to derive the coefficients. Compared with conventional linear prediction (LP) analysis [21], PLP analysis is claimed more consistent with human hearing.

Two additional preprocessing techniques may be introduced towards better feature modeling. First, delta and delta-delta coefficients are appended to the original feature vector, where deltas are the differential between successive frames. Second, linear discriminant analysis (LDA) can be applied to maximize the separation between target classes [22].

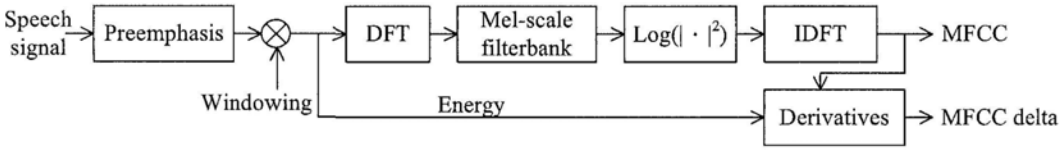


Figure 2.1: *Extraction of acoustic features, adopted from [1]*

2.1.2 Feature normalization and adaptation

Feature normalization aims at removing the bias caused by unrelated factors such as speaker, channel and environment. For example, vocal tract length normalization (VTLN) is a frequency warping technique that tries to normalize the inter-speaker difference due to the shift in formant centre frequencies with different vocal tract length [23]. Cepstral mean substraction (CMS) subtracts the mean cepstral value from each feature vector. Relative spectra (RASTA) filtering is a technique originally invented for predictive linear prediction (PLP) coefficients to remove slow channel variation. It can also be applied to cepstral features [24]. CMS requires the computation of long-term cepstral mean, while RASTA requires only a single-pass computation over the input data. The impact of RASTA on the LID performance was reported to be identical to CMS [2].

Feature compensation and adaptation serve similar purposes of feature normalization. They compensate variability in channel, session and/or other undesirable factors. Nevertheless, they are normally carried out after some preliminary processing in the feature domain, or in the model domain after a first-pass training. Examples for feature compensation include feature domain latent factor analysis (fLFA) and nuisance attribute projection (fNAP) [25]. Examples for adaption in the model domain include maximum-a-posteriori (MAP) adaptation and maximum likelihood linear regression (MLLR) adaptation. Both adaptation methods make use of unlabeled data to update model parameters. MAP adaptation constrains on the contribution of unlabeled data. MLLR adaptation is an affine transformation [26]. Joint factor analysis (JFA) model is another method to tackle inter-session variation. It is popularly used in LID systems in recent years [27, 28, 29].

2.2 Acoustic approach

Acoustic LID systems focus on the direct modeling of short-time spectral features introduced in Section 2.1.1. Gaussian mixture models (GMM) are typically employed to model the language identity. In the recent decade, the acoustic approach to LID is further enhanced with various implementations of machine learning and pattern recognition algorithms. For instance, **Maximum mutual information** and **support vector machine** make use of discriminative training algorithms. **Shifted delta cepstra** compensate the lack of long-range modeling in the acoustic approach.

2.2.1 Gaussian mixture modeling classification

In this method, a multi-variate Gaussian mixture model (GMM) models the distribution of short-time feature vectors from the same language. Let $\boldsymbol{v}_t$ denote the feature vector. It lies on a certain point in the feature space. Under the GMM assumption, its probability density equals a weighted sum of K normally distributed densities,

$$p(\boldsymbol{v}_t|\Theta) = \sum_{k=1}^{K} w_k b_k(\boldsymbol{v}_t). \tag{2.1}$$

In the above equation, $\Theta = \{w_k, \mu_k, \Sigma_k\}$ is the set of model parameters. $b_k(\boldsymbol{v}_t)$ is the observation probability specific to mixture k. It is computed using the mean statistics μ_k and the covariance statistics Σ_k. The probability terms are summed with mixture weights w_k. For a language n_t, Θ_{n_t} is constructed. Its parameters are found with training data from the language.

In recognition, the test utterance is transformed into sequence of short-time feature vectors, $\langle \boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_t, \ldots, \boldsymbol{v}_{T-1} \rangle$. The log likelihood that the sequence belongs to language n_t is computed by the following equation,

$$\mathcal{L}(\langle \boldsymbol{v}_0, \ldots, \boldsymbol{v}_{T-1} \rangle | \Theta_{n_t}) = \sum_{t=0}^{T-1} \log p(\boldsymbol{v}_t | \Theta_{n_t}), \tag{2.2}$$

which is essentially the joint probability of all short-time feature vectors with independence assumption. In the decision stage, language with the maximum log likelihood is chosen.

$$\hat{n}_t = \underset{n_t}{\operatorname{argmax}} \mathcal{L}(\langle \boldsymbol{v}_0, \ldots, \boldsymbol{v}_{T-1} \rangle | \Theta_{n_t}) \tag{2.3}$$

There are various techniques for the training of model parameters Θ_{n_t}. We label all training utterances belonging to language n_t with indices $\langle 0, 1, \ldots, r, \ldots, R-1\rangle$. The log likelihood for a training utterance r is $\mathcal{L}(\boldsymbol{v}[r]|\Theta_{n_t})$. The maximum likelihood (ML) criterion is widely adopted for the training of Θ_{n_t},

$$\mathcal{F}_{ML}(\Theta_{n_t}) = \sum_{r=0}^{R-1} \log p(\boldsymbol{v}[r]|\Theta_{n_t}), \tag{2.4}$$

Expectation maximization (EM) algorithm is used for the optimization of Θ_{n_t}, subject to maximum likelihood in Eq.(2.4) [30].

2.2.2 Improved methods for acoustic approach

Maximum mutual information

Maximum mutual information (MMI) is an example for the use of discriminative training in LID. With MMI, optimization is performed on posterior probability [31]. Suppose a model Θ_{n_t} is trained for target language n_t. The objective function of MMI is,

$$\mathcal{F}_{MMI}(\Theta_{n_t}) = \sum_{r=0}^{R-1} \frac{p(\boldsymbol{v}[r]|\Theta_{n_t})p(\Theta_{n_t})}{\sum_{\Theta_n} p(\boldsymbol{v}[r]|\Theta_n)p(\Theta_n)} \tag{2.5}$$

Apart from the true class log likelihood, in the optimization with MMI criterion the log likelihood of *competing classes* is also taken into accounts. Other discriminative training approaches include *minimum classification error* (MCE)[32], *minimum phone/word error* (MPE/MWE)[33] and *functional MPE* (fMPE)[34]. Compared with the Bayesian learning approach, these approaches involve additional considerations in the competing scores, the decision rule and/or the error in the course of modeling.

Support vector machines

Support vector machine (SVM) is a large margin method which also falls under the discriminative training approaches. SVM differs from other discriminative training approaches in its vector-based classifier backend. In the course of modeling, a kernel function is used to transform the input sequence to a fixed-length feature vector for vector-based learning. In this section, two kernel functions - generalized linear discriminant sequence (GLDS) and Gaussian supervector (GSV) are introduced.

Generalized linear discriminant sequence (GLDS) kernel [35] expands the feature vector by taking monomials. For instance, expanding a 38-dimensional MFCC plus delta feature vector up to degree 3 results in a feature space with a dimension of $C_0^3 C_0^{38} + C_1^3 C_1^{38} + C_2^3 C_2^{38} + C_3^3 C_3^{38} = 10660$.

The idea of Gaussian supervector (GSV) is to adapt a universal background model (UBM) GMM on a per utterance basis and then use the resulting shift to predict the language class [36, 37]. Dimension of GSV is $P \times K$ where K is the number of Gaussian mixtures in the UBM GMM and P is the dimension of feature vector.

Shifted delta cepstra

Shifted delta cepstra (SDC) modeling is an approach which endeavours to capture the dynamic properties of speech in the hope of better modeling the change of vocal tract in the course of articulation [38]. As its name goes, the major characteristic of SDC is the use of delta features between successive frames. Standard SDC configurations are specified by four parameters (N,d,p,k), where N is the number of cepstral coefficients obtained from a short-time acoustic frame; d is the advance and delay for the delta coefficients; p and k specify respectively the separation and the number of the blocks to concatenate together. In language recognition experiments, a standard configuration is 7-1-3-7 with the static cepstra prepended, producing a 56-dimensional SDC vector.

2.3 Phonotactic approach

In the *phonotactic* approach, LID systems model the co-occurrence statistics of sequences of allowable phones and phonemes in different languages. A typical phonotactic LID system has three components [39]. First, a *voice tokenizer* temporally groups the continuous flow of incoming voice into segment units and performs categorization. An automatic speech recognizer (ASR) is often applied to serve this purpose. Second, a *statistical language model* captures the language-dependent phonetic and phonotactic information of the tokenized units. Finally, a *language classifier* identifies the language of the test speech. The three components can be implemented with different input features and different modeling methods. A classical example is the parallel phone recognition followed by language modeling (PPRLM) [2]. With large margin training with support vector machines (SVM), in recent years the *bag of sounds* approach is used in tokenized system for speaker and language recognition [4, 40].

Parallel phone recognition followed by language model (PPRLM) is one of the prevailing methods for phonotactic LID since it was proposed in the nineties [2]. The structure of PPRLM is illustrated in Figure 2.2. Suppose we have three target-languages, Farsi, French and Tamil, in language recognition. First, multiple phone decoders recognize the test speech just like ordinary ASR engines do. These decoders do not have to be trained on the target languages. It depends of the availability of labeled speech. In our example, the three decoders form three parallel frontends. In every frontend output, multiple backend n-gram language models are trained for all target languages, resulting a total of 9 n-gram models. The output from all backends are averaged to give an overall language likelihood score.

Various advancements in phonotactic LID systems are observed in recent years. Gauvain proposed to consider multiple hypotheses in the PPRLM system

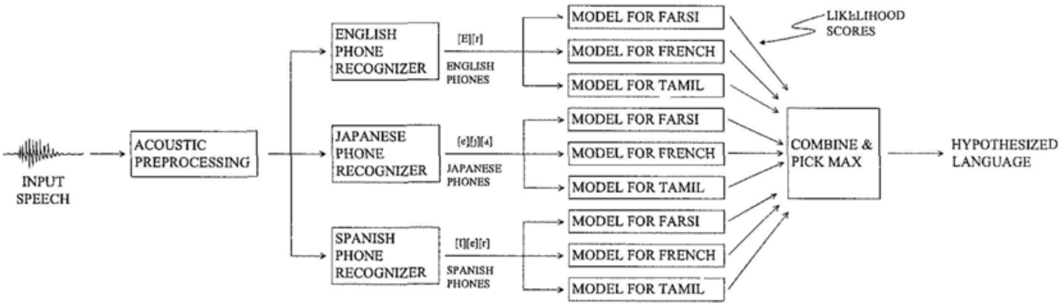


Figure 2.2: *PPRLM block diagram adopted from [2] with target languages being Farsi, French and Tamil*

with a lattice [41]. Recently, discriminative training of language recognizer, particularly with support vector machine (SVM), was proposed [4, 40]. Speech data is transformed to fixed-length feature vectors for vector-based training and testing. Usually the feature vectors represent the occurrence statistics of different tokens or token n-grams.

2.4 Prosodic features for LID

Prosody refers to the rhythmic and intonational properties in speech. The use of prosodic features for LID originates from some perceptual studies. In 1968, Atkinson suggested that human subjects can discriminate between English and Spanish poetry with intonation and duration cues [7]. In another study [11], syllabic rhythm was shown to be sufficient for an English-Japanese spoken language discrimination task by French-speaking listeners. Muthusamy *et al.* found some prosodic cues used by human listeners to distinguish Mandarin, Japanese and Vietnamese from other languages [42].

The three major types of prosodic properties in speech are duration, fundamental frequency (F0) and intensity. Unlike short-time acoustic features, these properties are observed over a relatively long time span [10]. Before prosodic features can be used for LID, there are a series of questions to answer. These questions include the time span of the linguistic unit for feature extraction, the actual features to extract, the way of modeling, etc.

2.4.1 Syllable: The basic unit for prosody

Prosodic features are suprasegmental. For many empirical tasks related to speech recognition using prosodic features, the basic unit for prosodic feature extraction is syllable, pseudosyllable or a suprasegmental unit of similar size [12, 43, 44, 45, 46]. Considering also some linguistic studies on prosodic properties [47, 48, 49, 50, 51, 52, 53], some principles on processing the acoustic signal on the syllable level for prosody modeling are deduced:

- The discussions of prosodic properties in linguistics often involve different levels of abstraction. This thesis studies the use of prosodic features in a computer-based application of LID. Focus will be put on the acoustic correlates of prosodic events.

- Linguistic analysis of prosodic properties often starts at the level of syllables and extends to words and phrases. Meanwhile, the complexity of an automatic speech processing algorithm may be too high for processing long phrases. To balance between the two, syllable is chosen to be the basic unit for prosodic feature extraction.

- Because there is not a consistent and language-independent definition on syllable in the acoustic domain of speech [12, 54], automatic segmentation is used to construct units called *pseudosyllables*. A pseudosyllable is a unit inferred from the results of automatic processing of speech, spanning over one linguistic segment and may not map exactly to the linguistic definition of a syllable [12, 55, 56].

- The surface representations of stress and tones are often anchored to vowels. (Pseudo)syllable alignments shall coincide with this anchor. In other words, syllabification shall be implemented through vowel detection. Instead of syllable boundaries, syllable nuclei are to be marked.

In this thesis, pseudosyllable is defined as the basic unit for prosodic feature extractions. A syllabification process to construct these units is necessary. There are different approaches for syllabification, and they can be classified into two categories. In the first category, syllabification is based on the phone alignments from an automatic speech recognition (ASR) engine [12, 55, 56, 57, 58, 59]. In the second category, an ASR engine is not used [46, 60, 61].

The ASR approaches for syllabification essentially group the ASR-generated phones to construct a pseudosyllable. The most common pseudosyllabic structure is $\mathrm{C}^{\mathrm{n}}\mathrm{V}$, where C and V refer to a consonant and a vowel respectively, n is a non-negative integer. With this approach, *pseudosyllable* is a unit having one or more consonants followed by a vowel, according to the results from an

ASR engine. Instead of training the ASR engine with precise phone models, it is usually adequate to merge phone classes to create a consonant(C)/vowel(V) binary set. From the C/V alignments, pseudosyllable boundaries are marked.

For the non-ASR approaches, other suprasegmental landmarks are used. For instance, pseudosyllables can be found by delimiting the pitch segment by the minimum point of a smoothed F0 contour [60]. In [46], excitation source information was used to find vowel onset points as the landmarks of pseudosyllables. Adami and Hermansky used inflection points and start or end of voicing to segment speech signal [61]. As such, the duration of a *pseudosyllable* will depend on the exact implementation method in an experiment.

2.4.2 Prosodic features

The three major prosodic feature types are duration, fundamental frequency (F0) and intensity. Various studies investigated the use of prosodic features for LID and other related speech and speaker recognition tasks. Because of the lack of a conventional set of prosodic features on a par with the short-time acoustic features, these studies differ in terms of the features and experiments involved.

Table 2.1 summarizes the prosodic features that were used for the various tasks. In almost all cases, the basic unit for prosodic feature extraction is a syllable-like unit. From the table, we can see combinations of different attributes under the three major prosodic feature types - duration, F0 and intensity.

Usage of specific features

Rouas *et al.* [10], Yin *et al.* [57] and Timoshenko and Höge [58] proposed to solve the LID problem by using speech rhythms only. Rouas [10] assumed a pseudosyllabic structure of $\mathrm{C}^{\mathrm{n}}\mathrm{V}$, and used a three-dimension feature vector to represent the rhythmic information of a pseudosyllable,

$$[D_{C1}+D_{C2}+\ldots+D_{C\mathrm{n}} \quad D_v \quad N_C]. \tag{2.6}$$

The first term, $D_{C1} + \ldots + D_{C\mathrm{n}}$, is the total duration of the consonantal

Table 2.1: *Comparison of the prosodic features used for different tasks*

Author Task description	Prosodic features used	Backend classifier
Rouas *et al* [10]* Language identification	Total consonant cluster duration, total vowel duration, Complexity of consonantal cluster	GMM
Rouas [12]*¶ Language identification	Phrase F0 curve gradient, pseudosyllable F0 residue, energy contour gradient in pseudosyllables, duration of vocalic segments	GMM
Rouas *et al* [62]* Language identification	Total duration of consonantal segments and vowel segments Number of segments in the consonantal clusters F0 mean, F0 variance, F0 skewness, F0 kurtosis Accent location, F0 bandwidth	GMM
Yin *et al* [57]*¶ Language identification	Duration of segments assuming unvoiced-voiced sequence in speech	VQ
Timoshenko and Hoge [58] Language identification	Duration of pseudosyllables, assuming C^nV structure	ANN
Lin and Wang [60]* Language identification	Legendre polynomial approximation to a segment of pitch contour	GMM
Piat *et al* [63]*¶‡ Accent identification	Ratio of syllable duration over word duration, Average energy & energy profile of each syllable, F0 slope, speaker-normalized F0	HMM
Levow[64]*¶♯ Tone and accent learning	Mean pitch across a syllable, Pitch slope in syllable final, Extended features Maximum and mean pitch from adjacent syllables, Difference features Changes in intensity maximum, pitch maximum, pitch mean, mid-point, slope	Asymmetric k lines clustering
Kochanski *et al* [59] Speech/Poem analysis	Segment(C/V) duration, loudness, degree of aperiodicity early/late loudness in segment, spectral change	Coefficient of determination
Biadsy and Hirschberg[55]*¶ Dialect identification	mean F0, pitch slope, pitch peak alignment, RMS intensity, duration, delta between two pseudosyllables	HMM
Mary and Yegnanarayana [46]* Speaker/Language Recognition	Change in F0, F0 peak from voice onset point(VOP), amplitude tilt, duration tilt, distance between successive VOPs, duration of voiced region, delta of log energy between pseudosyllables	Neural network
Peng and Wang [65]*¶§ Tone recognition	F0 values taken in multiple time points in a syllable, duration of F0 contour, subsyllabic mean of log-energy, F0 and energy values in adjacent syllables	SVM
Hazen and Zue [66] Language identification	F0, delta F0, segment duration	GMM
Thymé-Gobbel and Hutchins [43]*◇ Language identification	Pitch contour shape, delta max pitch / delta mid-point pitch between syllables, distance between syllables, syllable duration, delta duration, amplitude contour shape, delta mid-point amplitude / delta max amplitude between syllables, low frequency FFT of amplitude envelop, syllable location & speaking rate within a breath group	Histogram comparison
Shriberg *et al* [56]* Speaker recognition	F0, energy and duration features in various extraction & normalization methods with syllable-NERF modeling	SVM

*Features are extracted within one syllable, pseudosyllable, or similar supra-segmental unit

¶Normalization measures applied to some features

♯Per-speaker z-score normalized log-scale values are used for all features

‡Energy is subtracted by the utterance maximum, 1st,2nd derivatives over the contour are also taken

§Moving window normalization is applied to F0 and energy values

◇Averages, deltas, standard deviations measures are taken Individual features are combined into feature pairs Correlations of measures are taken cluster in a pseudosyllable. The second term, D_v, is the duration of the vowel segment. The last term, N_C, is the number of segments in the consonantal cluster, indicating the complexity of the pseudosyllable. In the studies of Yin *et al.* and Timoshenko and Höge, duration attributes are represented by even simpler features with one or two dimensions [57, 58].

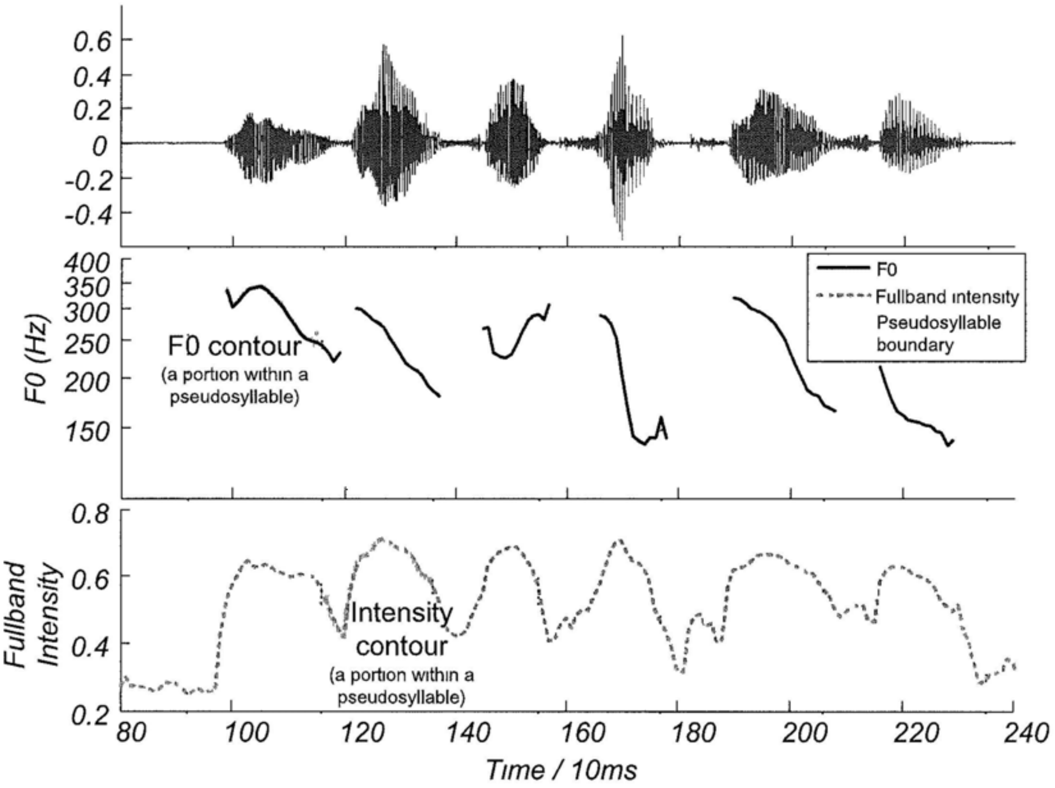


Figure 2.3: *F0 and intensity contours and portions within a pseudosyllable*

Fundamental frequency (F0) and intensity exhibit themselves in the form of short-time measurement sequences. These sequences are commonly known as *contours*. Figure 2.3 shows the F0 and intensity contours. The top panel of the figure is the speech waveform spoken in Mandarin. The vertical dotted lines mark the boundaries of six pseudosyllables. They are generated with an ASR approach for syllabification. The middle and bottom panel show the F0 contour and the intensity contour respectively. Pseudosyllable-based attributes are extracted from the portion of contours within the corresponding syllable boundaries.

Among different F0 attributes, the typical ones include F0 mean, F0 gradient and delta attributes. F0 mean is the average value of all frame-based measurements from the portion of F0 contour. F0 gradient is obtained by first performing linear regression of the portion of F0 contour, and then measuring the slope of the regression line. In [60], Lin and Wang extended this idea and applied Legendre polynomial approximation. Delta features refer to the difference of a particular F0 attribute across two contiguous pseudosyllables. F0 attributes are found to be widely used in different studies as reported in Table 2.1.

Intensity attributes are derived in a similar manner to the F0 attributes. Typical intensity attributes include average and delta amplitude. Many intensity attributes have a large dynamic range, sometimes logarithm is taken [56, 63].

Usage of a large feature set

In Thymé-Gobbel and Hutchins' study [43], The "discrim" system investigated 224 individual prosodic features, including pitch, syllable duration, shape of amplitude contour in terms of average, delta, standard deviation. Individual features were also combined into feature pairs. Nevertheless, among these 224 attributes only a small subset were used in training and discrimination modules [43].

Shriberg studied a large set of prosodic features, known as *syllable non-uniform extraction region features* (SNERF), for speaker recognition [56]. SNERF are defined both by the *region* from which the features are extracted, and by the type of *features* extracted within that region [67]. In principle, the extraction *region* of SNERF covers a syllable. An ASR engine with a typical English phone set is trained, and a program called "tsylb2" uses a set of hand-crafted rules to match the English phones to English syllables [56].

Following the trichotomy of prosodic features, SNERF include pitch, energy and duration [56]. For pitch features, multiple variants of frame-based F0 measurements are obtained. Examples are maximum, mean, minimum values in the F0 measurements. Probability of pitch halving/doubling in each frame is also calculated, by using a log-normal tied-mixture model of pitch. The extraction of energy features in SNERF is similar to that of pitch features. The only difference is that extraction regions for energy features are not limited to voiced frames. For duration, multiple extraction *regions* are used to extract features. These include syllables, and subsyllabic units like onset, nucleus and coda.

SNERF are quantized features. By repeating different quantization resolutions from 2 up to 60, several versions for each quantized feature are generated. Feature normalization is another issue. Most syllable-based SNERFs are normalized over a longer window covering multiple units. This normalization approach captures the *syntagmatic* properties of the features, where the contrast of prosodic properties can be made with reference to the neighbouring speech units. Normalization is carried out by dividing by/subtracting mean, z-normalization (subtracting mean, then divide by standard deviation) or finding percentiles.

2.5 Language classification with SVM

A *language classifier* is an important module in an LID system which identifies the language of the test speech. All analogous modules in related speech recognition tasks with prosodic features are summarized under the column of *backend classifier* in Table 2.1. The major classifiers used include Gaussian mixture model (GMM), hidden Markov model (HMM) and support vector machine (SVM). In this thesis, SVM will be implemented to serve language classification purpose. SVM is a large margin training method originating from a statistical learning problem for binary classification. Traditional classifiers like the Bayesian network model the probability distributions of the training data in every target class separately [68]. Overfitting may occur and the generalization ability is degraded. In a binary classification problem, SVM is trained in a discriminative manner. The training algorithm finds a decision function with maximum generalization ability [69]. SVM is widely used in various applications in speech processing, speaker and language recognition

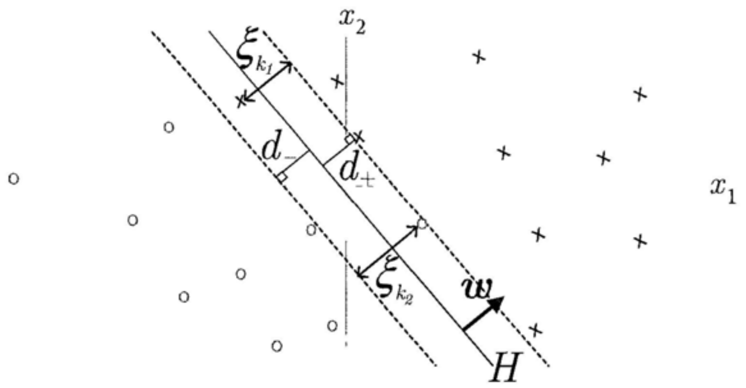


Figure 2.4: *The binary-class training data, hyperplane and the margin*

[4, 40, 65, 68, 70, 71, 72, 73, 74].

In each LID trial, a particular language is detected. To accomplish this goal, a binary classifier is trained from a multi-lingual training database with many utterances. After some transformation, each utterance is represented by a P-dimensional feature vector $\boldsymbol{x}$. We want to distinguish those feature vectors $\boldsymbol{x}$ which belong to the target language against all other $\boldsymbol{x}$. Figure 2.4 is a simplified illustration with some training samples in a two-dimensional feature space. In the figure, training samples of the two classes are marked by crosses and circles respectively. The training procedure finds an optimal hyperplane H which separates these two classes of $\boldsymbol{x}$ subject to a maximum *margin*, $|d_+|+|d_-|$. In the following, we consider the three elements in SVM training, which are **linear separability** of data, the **margin** and the **slack variable**. From these three elements, the objective function of SVM is formulated.

Linear separability

In a P-dimensional space ($\mathbb{R}^P$), a hyperplane H can be defined by the following equation,

$$H = \{\boldsymbol{x} \in \mathbb{R}^P | \boldsymbol{w}^T\boldsymbol{x} + b = 0, \boldsymbol{w} \in \mathbb{R}^P, b \in \mathbb{R}\}. \tag{2.7}$$

$\boldsymbol{w}$ and b can be referred to as *weights* and *bias* respectively [75]. *Weight* is a normal vector orthogonal to H. Its value affects the orientation of H. *Bias* is the scaled offsets of H from the origin. It translates the position of H.

The prerequisite for an optimal hyperplane H is that H separates the training samples of the two classes by two different halfspaces. This constraint is referred to as *linear separability*. Let y_k be the class label of the k^{th} sample. It takes either of the values ±1. Also let $\boldsymbol{x}_k$ be the feature vector of the k^{th} sample. The linear separability constraint can be described by an inequality [76],

$$y_k \left(\boldsymbol{w}^T \boldsymbol{x}_k + b \right) \geq 1 \quad \forall k. \tag{2.8}$$

Margin

Recall that the training procedure finds the optimal hyperplane H. Besides *linear separability*, an optimal H has a large *margin*. *Margin* is the sum of two distances, $|d_+|$ and $|d_-|$, which are measured from H to the closest sample in each class. $|d_+|$ and $|d_-|$ are illustrated in Figure 2.4. We use the normal form of H in Eq.(2.7). With some derivations, the value of *margin* can be expressed in terms of $\boldsymbol{w}$ [76],

$$|d_+| + |d_-| = \frac{2}{\|\boldsymbol{w}\|}. \tag{2.9}$$

Slack variables

Figure 2.4 illustrates the case when linear separability cannot be attained. We have to neglect some data points such that the remaining binary-class samples become linear separable again. In the figure, two samples (k_1 and k_2) are neglected. For each neglected sample, linear separability constraint (Eq.(2.8)) is relaxed by introducing a slack variable ξ. The modified constraint equation becomes

$$y_k \left(\boldsymbol{w}^T \boldsymbol{x}_k + b \right) \geq 1 - \xi_k ; \xi_k \geq 0 \quad \forall k. \tag{2.10}$$

The scenario illustrated in Figure 2.4 is modeled by $\xi_{k_1} > 0$ and $\xi_{k_2} > 0$. For other samples, no relaxation by slack variables is needed and the corresponding ξ's vanish. The margin derived in this way is no longer the absolute separation between two classes of samples. It is referred to as a *soft margin*. With more relaxation, a larger value of the soft margin could be returned, which seems to indicate good bisection by H. Nevertheless, more relaxation also means more samples (such as k_1 and k_2 in Figure 2.4) are disregarded, and a penalty is incurred by this act of negligence. A common penalty function is:

$$C \sum_{k=1}^{K} (\xi_k)^m \tag{2.11}$$

C and m are penalty function parameters. In this thesis, we assume C is any positive constant and $m = 1$.

Optimization objective

In SVM training, the value of the margin (Eq.(2.9)) is maximized (or its reciprocal is minimized). Also, the penalty incurred by linear separability relaxation (Eq.(2.11)) is minimized. Written in vector form, the whole optimization problem is formulated as,

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\boldsymbol{\xi} & \\ \text{subject to(s.t.)} \quad \mathbf{diag}(\boldsymbol{y})(\boldsymbol{X}^T\boldsymbol{w} + b) - \mathbf{1} + \boldsymbol{\xi} &\succeq 0, \\ \boldsymbol{\xi} &\succeq 0. \end{aligned} \tag{2.12}$$

$\boldsymbol{w} \in \mathbb{R}^P$ is the normal vector orthogonal to the separating hyperplane H. $\boldsymbol{X} \in \mathbb{R}^{P\times K}$ is the collection of K observation vectors in the P-dimensional input space. $\boldsymbol{y} \in \mathbb{R}^K$ is the class label of the training data. $\mathbf{diag}(\boldsymbol{y})$ is a diagonal matrix with elements $y_1,\ldots,y_K$. $\mathbf{1}$ is a vector with K elements of all ones. $\boldsymbol{\xi} \in \mathbb{R}^K$ is a vector of K non-negative slack variables which relax the separability constraint.

The whole problem is convex and can be solved by quadratic programming. To avoid the high problem dimension, it is popular to transform the original primal problem to a dual problem. Optimality conditions in the primal and the dual problems are well defined. In practical implementation of support vector machine training, other issues have to be tackled. For example, the methods to map the input samples to high-dimensional space vary [76]. There are different choices of penalty functions (Eq.(2.11)) when the linear separability constraint is relaxed. Finally, when the problem dimension is still large after transformation to the dual problem, the data have to be decomposed into many working sets and optimization is done iteratively [77, 78, 79].

Chapter 3

Acoustic tokenization

A prosody-based LID system is proposed in this thesis. Because prosodic features are suprasegmental and are extracted from a rather long linguistic unit in the syllable level, the prosody-based LID system works in an analogous way as a phonotactic LID system introduced in Chapter 2 does. The general flow of such a system is summarized by figure 3.1. An *acoustic tokenizer* frontend segments the speech into syllables, from which prosodic features are extracted. In the *statistical language modeling* module, occurrence statistics of the prosodic features are computed, using the *bag of sounds* approach, for example. Finally, the statistics are used to train a *language classifier*.

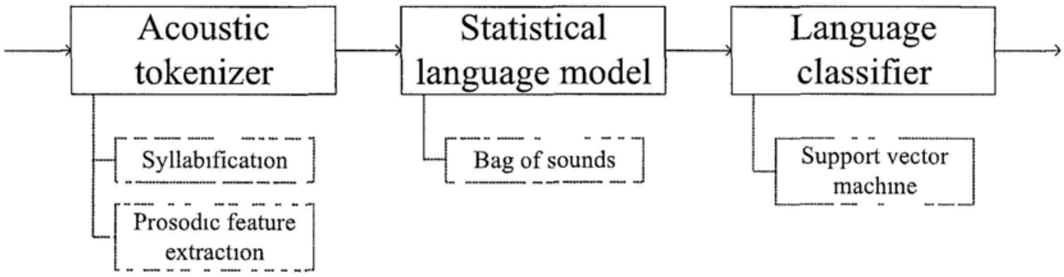


Figure 3.1: *A phonotactic LID approach for prosodic features*

In this chapter, our implementation of the *acoustic tokenization* frontend in the LID system is explained in detail. The tokenization frontend temporally groups the continuous flow of incoming voice into segment units and prosodic features are extracted. According to the principles described in Section 2.4.1, the segment unit to use is *pseudosyllable*. The **syllabification algorithm** is

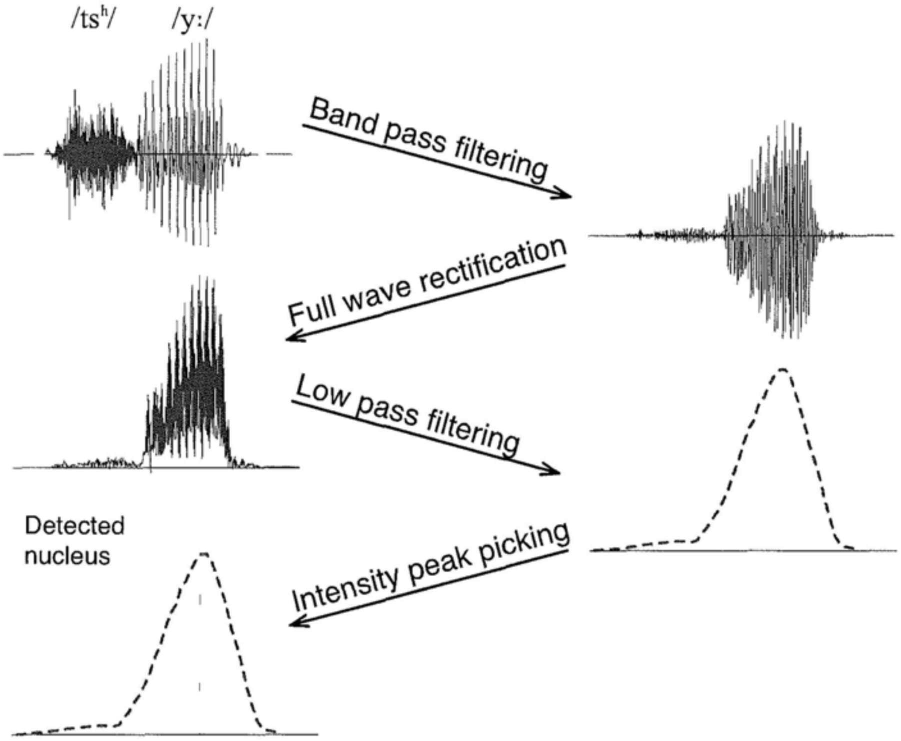


Figure 3.2: *Detecting the syllable nucleus of /ts^{h}y:/*

introduced in Section 3.1 and its effectiveness are justified by **syllabification experiments** in Section 3.2. From each pseudosyllable, a large number of **prosodic attributes** will be extracted. These attributes will be introduced in Section 3.3.

3.1 Syllabification algorithm

We implement a syllabification algorithm which segments speech by finding the nuclei of pseudosyllables. With our intuitions on syllables, the detection to pseudosyllabic nuclei relies on the loudness of speech. At nucleus positions, it is assumed high acoustic intensity must exist because of mouth opening.

In the syllabification algorithm, we extract the intensity profile in the sonorant band by **band pass filtering**. Local peaks are detected using **rectification** and **low pass filtering**. These peaks are considered as candidates of the nuclei of pseudosyllables. All candidates are evaluated by a scoring function in the **intensity peak picking** part. Eligible peaks are accepted as a pseudosyllabic nucleus. This algorithm is a non-ASR approach to syllabification. It is referred to as *peak-picking syllabification* (PPS) hereinafter.

Figure 3.2 shows an example of PPS with a Cantonese syllable /ts^{h}y:/ (with the character 處). The syllable has an aspirated affricate onset and a long high front rounded vowel. Band pass filtering removes the consonantal contents. Low pass filtering returns only one peak in the vocalic region, whose eligibility is confirmed by the scoring function. The detected nucleus is marked in the figure.

3.1.1 Band pass filtering

In the first step of PPS, a band pass filter is used to reduce the non-vocalic components of speech, as well as background noise. The considerations in this step include the passing band and the type of the band pass filter. In an earlier study, Weigel used a relatively broad frequency band (250Hz to 2500Hz) for the detection of vocalic components in speech [80]. Mermelstein used the telephone bandwidth from 500Hz to 4000Hz [81].

There were also studies in which a narrower passing band was used. Pfitzinger *et al.* found the frequency regions from 360Hz to 1650Hz useful [54]. Howitt tried different band edges within the first formant (F1) frequency bands and found that passing frequency from 0Hz to 650Hz produced the optimal results [82]. Pellegrino and Andre-Obrecht used the short-time energy from the Mel-scale frequency filters in the range of 100Hz to 1000Hz to locate voiced speech region [83]. The passing band is called sonorant band, which corresponds to the frequency regions where vocalic segments best demonstrate their energy concentration. In the ideal case, the filtered signal contains only the vocalic component of speech.

In this thesis, a digital filter with passing band of 300Hz to 1000Hz will be used. This band is expected to exclude the nasal sounds in low frequency regions and the formants in high frequency regions, which are zero-crossing frequencies related to the phone type and quality. The digital filter used is a fourth-order elliptic filter with 0.5dB passband ripple and 70dB stopband rejection.

3.1.2 Full wave rectification and low pass filtering

The vocalic component of speech can be regarded as a simple carrier modulated by the intensity profile. In the second and third steps of PPS, full wave rectification and low pass filtering act as a standard demodulator. The extracted temporal envelop will be regarded as intensity profile.

An illustrative example is shown in Figure 3.3. The vocalic component is shown in the left sub-figure. The effect of full wave rectification is illustrated in the right sub-figure. It shifts the formants upward by doubling their frequencies without changing the shape of the envelope. Finally, a low pass filter outputs the temporal envelope. The low pass filter is implemented as moving-window average, i.e.

$$y_{LPF}[n] = \frac{1}{T} \sum_{\tau=-T/2}^{T/2} x_{LPF}[n+\tau], \qquad (3.1)$$

where x_{LPF} is the input and y_{LPF} is the output of the filter. The frequency response of this linear filter is a sinc function with the mainlobe bandwidth of $\frac{1}{T}$Hz. In this study, the mainlobe of the linear filter is set to be 20Hz wide, and the width of moving window is 50ms.

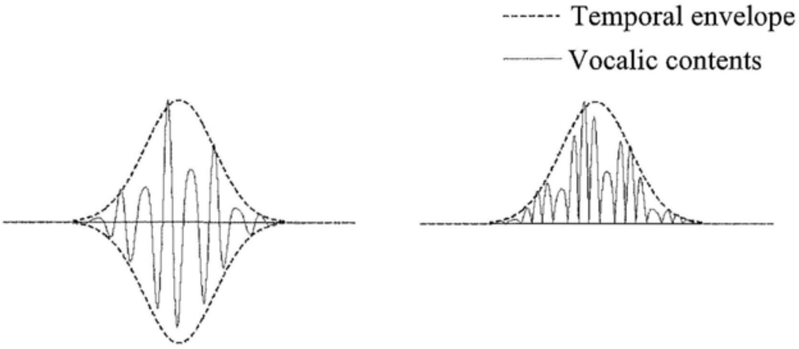


Figure 3.3: *Full wave rectification of the vocalic component of speech for nuclei detection. (left) Carrier of vocalic content modulated by the intensity profile (right) Full wave rectified signal for the demodulation of the intensity profile*

3.1.3 Intensity peak picking

In the final step of PPS, an algorithm detects pseudosyllabic nuclei from the intensity profile. Two procedures are involved. First, a moving window slides along time and determines the positions of local maxima and minima in the intensity profile. By varying the width of the moving window, more or fewer extrema points will be returned. This width has to be tried empirically. To prevent from false alarms due to the perturbations in the intensity profile, a median filter is used. A minimum syllable width is also imposed.

The second procedure of intensity peak picking is to verify whether a local maximum is a pseudosyllabic nucleus. Detected local maxima, or peaks, in the intensity profile are regarded as candidates of pseudosyllable nuclei. A scoring function comprising four criteria is proposed to quantify the eligibility of a peak for being a pseudosyllabic nucleus.

In the following, the four criteria and the scoring function will be introduced. The working principle of intensity peak picking is to have higher acceptance by the moving window first. Subsequently, the scoring function is used to eliminate the false alarms.

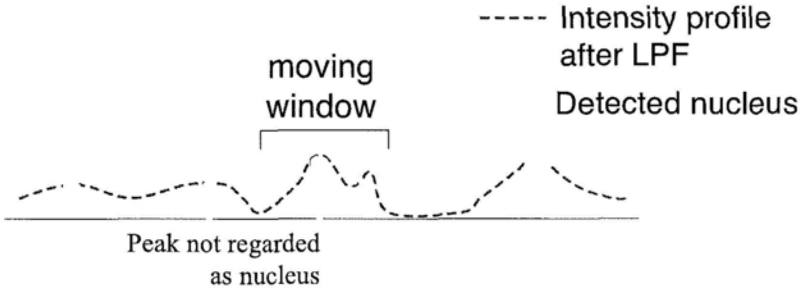

----- Intensity profile after LPF
Detected nucleus
moving window
Peak not regarded as nucleus

Figure 3.4: *Using a moving window, the peak picking algorithm rejects some local maxima in the intensity profile in syllabification*

Voicedness

Voicedness is a binary criterion, which means it contributes either one or zero to the scoring function. This criterion is proposed because pseudosyllable nuclei are assumed to lie on vowels. The binary decision of voiced/unvoiced is made by comparing the intensity of speech in the frequency bands from 300Hz to

1500Hz, with a voicing threshold. The threshold is determined for every speech utterance from 3 seconds to 30 seconds long with histogram analysis.

Peak height

Peak height is a continuous-valued criterion in the scoring function. The larger the peak height, the more probable this peak being a pseudosyllable nucleus. With the local maxima and minima detected by the moving window, *peak height* is defined as the offset of the peak amplitude with respect to the immediately left minimum point plus that with respect to the immediately right minimum point.

Peak range

Peak range is a continuous-valued criterion. It counts the number of times a particular peak is reported to be the local maximum by the moving window. A pseudosyllable nucleus is expected to exhibit maximum intensity, captured by the detection window at multiple positions. On the contrary, maximum signal amplitude caused by perturbations of signals, weak glides or nasals is generally local, and reported as maximum by the detection window at one or two positions only. Assume the temporal width and shift of the detection window is w_{PPS} and τ_{PPS} respectively, it is obvious that the upper bound of *peak range* is $\frac{w_{PPS}}{\tau_{PPS}}$. That means a candidate is reported to be a local maximum point whenever the moving detection window covers.

Stand-alone peak

A peak is said to be *stand-alone* if there are no other peaks in the vicinity. We define this vicinity as a region bounded by the nearest local minima on the left and right. We take an example of evaluating the third peak in Figure 3.4. The local minima immediately after the peak is caused by perturbations and thus is not considered as a valid minimum. This third and the immediately following peaks are not *stand-alone* peaks.

The *stand-alone peak* criterion marks naturally-occurred *stand-alone* peaks with a value one. In case there are more than one peak, it chooses the best peak so as to enforce the *stand-alone* criterion. The peak that scores the highest according to the criteria of *voicedness*, *peak height* and *peak range* will receive a value one. In Figure 3.4, the first, second and last peaks are naturally-occurred *stand-alone* peaks. The third peak is an enforced *stand-alone* peak, which is preferred to the one immediately following.

The scoring function

With the four criteria, an overall scoring function is computed to quantify the eligibility of a detected local maximum point to be a pseudosyllable nucleus.

$$\begin{gathered} Eligibility = {w_{PPS}}^T C_{PPS}, \\ \text{where } w_{PPS} = [w_1 w_2 w_3 w_4]^T; C_{PPS} = [C_1 C_2 C_3 C_4]^T. \end{gathered} \tag{3.2}$$

C_1,C_2,C_3 and C_4 are the four criteria of *voicedness*, *peak height*, *peak range* and *stand-alone peak* respectively, corresponding to the four weights w_1, w_2, w_3 and w_4. Criteria C_1 and C_4 are binary variables having values of either 0 or 1. Criteria C_2 and C_3 have their values normalized to a range between 0 and 1. Different weight combinations $[w_1,w_2,w_3,w_4]$ and detection thresholds are tried exhaustively for an optimal value for the eligibility scoring.

3.2 Experimental results in syllabification

Experiments are carried out with three read-speech corpora, namely TIMIT in English, CUSENT in Cantonese and 863 in Mandarin. The corpora contain multiple speakers, and the gender is balanced. Each speech utterance in these corpora is from 2 to 5 seconds long. The number of utterances in 863, CUSENT and TIMIT are 205, 247 and 204, respectively. The speech contents are from news articles or passages designed to meet the phonetic balance criterion. Broadband clean recordings are downsampled to telephone bandwidth

Table 3.1: *Experimental results on syllabification conditioned on the width of moving window and the weight factor*

Moving window	Corpus	w_{PPS}[†]	Detection threshold	Insertion	Deletion	Vowel error rate (VER)
80ms	863 (Mandarin)	[0.1 0.1 0.8 0.2]	1	6.61%	5.30%	11.91%
	CUSENT (Cantonese)	[0.1 0.2 0.5 0.5]	1	4.14%	2.12%	6.26%
	TIMIT (English)	[0.2 0.5 0.2 0.8]	1	4.26%	17.54%	21.80%
	Overall	[0.1 0.2 0.2 0.8]	1	6.24%	8.13%	14.36%
130ms	863 (Mandarin)	[0.1 0.5 0.2 0.8]	1	4.16%	4.71%	8.87%
	CUSENT (Cantonese)	[0.1 0.5 0.2 0.8]	1	2.05%	2.59%	4.64%
	TIMIT (English)	[0.1 0.8 0.2 0.8]	1	2.40%	20.48%	22.88%
	Overall	[0.1 0.5 0.2 0.8]	1	2.52%	9.89%	12.41%

[†] w_{PPS} are the weight vector $[w_1 w_2 w_3 w_4]$ corresponding to the four peak picking criteria (Eq.(3.2))

before the syllabification algorithm is applied.

To calculate the detection accuracy, the detected pseudosyllables are compared with some reference data. Phone-level forced alignment or automatic segmentation of sub-syllabic units are provided with the corpora. From these alignments, boundaries of reference syllables are generated. An assumption of $\mathrm{C^nV}$ reference syllable structure is enforced.

In the proposed algorithm, the width of the moving window has to be empirically determined. Two widths, 80ms and 130ms are considered. They roughly correspond to the pseudosyllable durations in English and Chinese speech respectively. According to Equation (3.2), different weights are applied to the four criteria. Four weights, 0.1, 0.2, 0.5 and 0.8, are freely combined to form many weight factors. Together with three proposed detection thresholds, 1, 1.5 and 2, an optimal combination of weights and threshold subject to a high syllabification accuracy is found by greedy search.

The performance of syllabification algorithm is reflected as insertions, deletions and vowel error rate (VER). Insertion rate measures the percentage of reference syllables where more than one pseudosyllable is detected by the algorithm. Deletion rate measures the percentage of reference syllables where there is a detection miss. Vowel error rate (VER) is the sum of insertions and deletions.

Inferring from the syllabification results, the three languages fall into two categories: Cantonese and Mandarin syllabification benefit from using a longer moving window (130 ms), while English syllabification benefits from using a shorter moving window (80 ms).

For Chinese, the vowel error rate (VER) is below 10% with the 130ms moving window. The four peak-picking criteria, in the descending order of importance, are *stand-alone peak*, *peak height*, *peak range* and *voicedness*. This rank of significance is preserved for English syllabification with the 80ms moving window, except that *voicedness* receives a slightly higher importance. English syllabification has a higher VER about 20%.

The deletion rate of English pseudosyllables is significantly higher than other error terms. In a stress language like English there will be some syllables whose intensity is too small to be detected. It is an open question whether the errors should really be considered as deletions, or the definition of syllables in these languages should be revised.

3.3 Extraction of prosodic attributes

There exists no standard prosodic feature set on a par with the cepstral features for automatic speech recognition. As a result, many different prosodic features were reported in previous studies (Section 2.4.2). Despite some commonality, the choices of features and the exact definitions of individual feature parameters are highly application- and task-dependent. The task of LID involves multiple languages, each of which has its distinctive properties in prosody. It is not reasonable to expect that a single type of features would be adequate to distinguish all of the languages. Effective LID may require the joint contributions of many individual features. Such an approach was shown successful in the speaker recognition task [56].

In this section, we introduce a comprehensive set of *prosodic attributes* that is potentially useful for LID. Here a *prosodic attribute* refers to an explicitly defined measurement from the acoustic signals . These attributes describe the suprasegmental variation of F0, intensity, and duration in many different ways. Referring to previous studies, in this thesis 105 prosodic attributes are explored. These

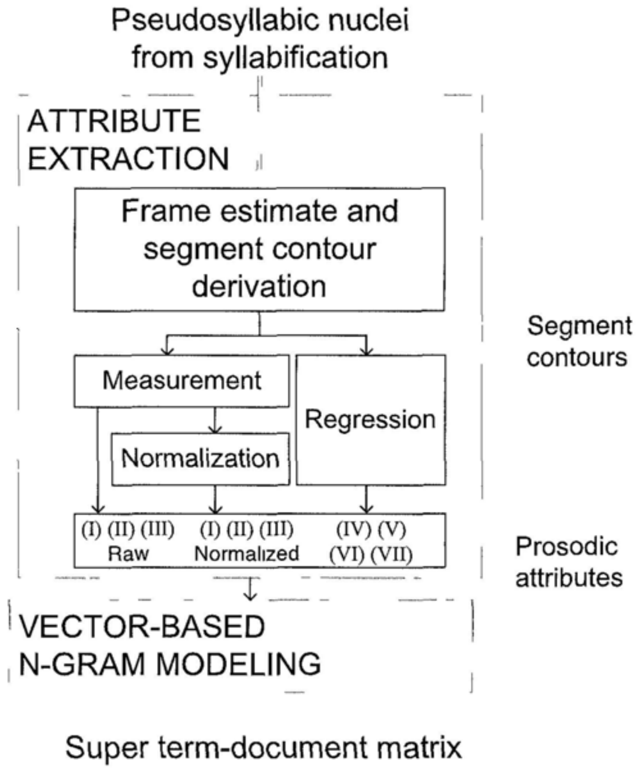

Pseudosyllabic nuclei from syllabification
ATTRIBUTE EXTRACTION
Frame estimate and segment contour derivation
Segment contours
Measurement
Regression
Normalization
(I) (II) (III) Raw
(I) (II) (III) Normalized
(IV) (V) (VI) (VII)
Prosodic attributes
VECTOR-BASED N-GRAM MODELING
Super term-document matrix

Figure 3.5: *Prosodic attribute extraction*

prosodic attributes are divided into seven groups. They are (I) **F0 basic**, (II) **Intensity basic**, (III) **Duration basic**, (IV) **F0 regression**, (V) **Intensity regression**, (VI) **F0 residue** and (VII) **Intensity residue**. Group (I),(II),(III) are prosodic features in basic forms. Within each group there is a number of attributes derived from different frame-based or syllable-based *measurements*, and with different *normalization* methods. Group (IV),(V),(VI),(VII) are related to the polynomial *regression* of F0/intensity contours. No normalization is needed.

Prosodic attribute extraction is a multi-stage process illustrated in Figure 3.5. First, the *frame estimates* of pitch and intensity are obtained. From these short-time estimates, *F0 segment contours* and *intensity segment contours* are derived. Then, some *measurements* in F0/intensity/duration are obtained in the *basic forms.* These measurements can be *frame-based* or *syllable-based measurements.* The final step is *normalization*, after which precisely defined prosodic attributes in Group (I),(II) and (III) are obtained. On the other hand, polynomial *regression* of the segment contours returns prosodic attributes in Group

(IV),(V),(VI) and (VII).

Apart from verbal descriptions, in the following a prosodic attribute is also denoted by $v_{(i,s)}$, where the subscript i is a shorthand indicating the nature of the prosodic attribute, and s is the index to the pseudosyllable. In this chapter's discussion on feature extraction, the pseudosyllable index s will be omitted for notation simplicity. v_i will be referring to the prosodic attribute i from a pseudosyllable not explicitly stated.

3.3.1 Frame estimates and segment contours

Frame estimates for F0, intensity and duration

The large number of F0, intensity and duration attributes originate from short-time *frame estimates.* The F0 attributes are from frame-based pitch estimates generated by a pitch extraction algorithm [56, 84]. Intensity attributes are computed from frame-based intensity estimates, which are represented by the RMS energy values [84]. There are no frame estimates for duration, but duration attributes can be inferred from the *segment contours* derived below.

Segment contour: definition

Short-time pitch and intensity estimates form discrete sequences. The typical time step (frame shift) between successive samples is 10ms. The sequences extend to continuous lines called *contours.* As prosodic attributes are extracted on the pseudosyllable level, we are interested in the corresponding portion of the contours (Figure 2.3). For the sake of clarity, such portions of contours are referred to as *F0 segment contour* and *intensity segment contour* hereinafter.

In later discussions, a *segment contour* will be denoted by $\boldsymbol{f}$. Suppose there are T samples in $\boldsymbol{f}$,

$$\boldsymbol{f} = [f[0], f[1], \ldots, f[T-1]]^T, \tag{3.3}$$

$f[t-1]$ denotes the t^{th} point in the *segment contour.* $\boldsymbol{f}$ should be taken from some pseudosyllable indexed s, but for simplicity, the pseudosyllable index is not shown. Moreover, the notation for an *F0 segment contour* is not distinguished from that of an *intensity segment contour*, unless ambiguity occurs.

Segment contour derivation

With the pseudosyllabic boundaries from an ASR approach for syllabification (Section 2.4.1), the *segment contours* are readily derived. In this study, syllabification is done with PPS. It only finds pseudosyllabic nucleus but not pseudosyllabic boundaries (Section 3.1). With only the locations of nuclei, we resolve to the shape and trajectory of the F0 contours to derive *segment contours.*

Figure 3.6 shows the derivation of *segment contours* under PPS. On the bottom panel, PPS detects the pseudosyllabic nuclei. Assuming the C^nV syllabic structure, and using the fact that F0 is often undefined in consonantal regions; thus, an *F0 segment contour* is constructed as the longest continuous extension of F0 contour from the pseudosyllabic nucleus. For a syllable with a glide or approximant onset, where the F0 contour becomes continuous across two pseudosyllables, the *segment contour* will be bisected at the point with the lowest sonorant band intensity.

After the *F0 segment contours* are defined, *intensity segment contours* are constructed to align with the F0 segment contours. The use of F0 characteristics to align *intensity segment contours* is a crude approximation. Nevertheless, it is also adopted in the construction of SNERF (Section 2.4.2) [56]. More investigations would be needed in the future.

Figure 3.6 shows a perfect case of *segment contour* derivation. The *segment contours* align well with the syllable alignments produced by the ASR engine on the top panel. The segment contour of each pseudosyllable has a finite duration, this can be used as a duration attribute.

In the subsequent prosodic attribute extraction process, contours in even longer temporal range will be considered. A *pair contour* refers to the contour spanning across two pseudosyllables. A *triplet contour* spans across three pseudosyllables. An *utterance contour* spans across a pause-delimited utterance.

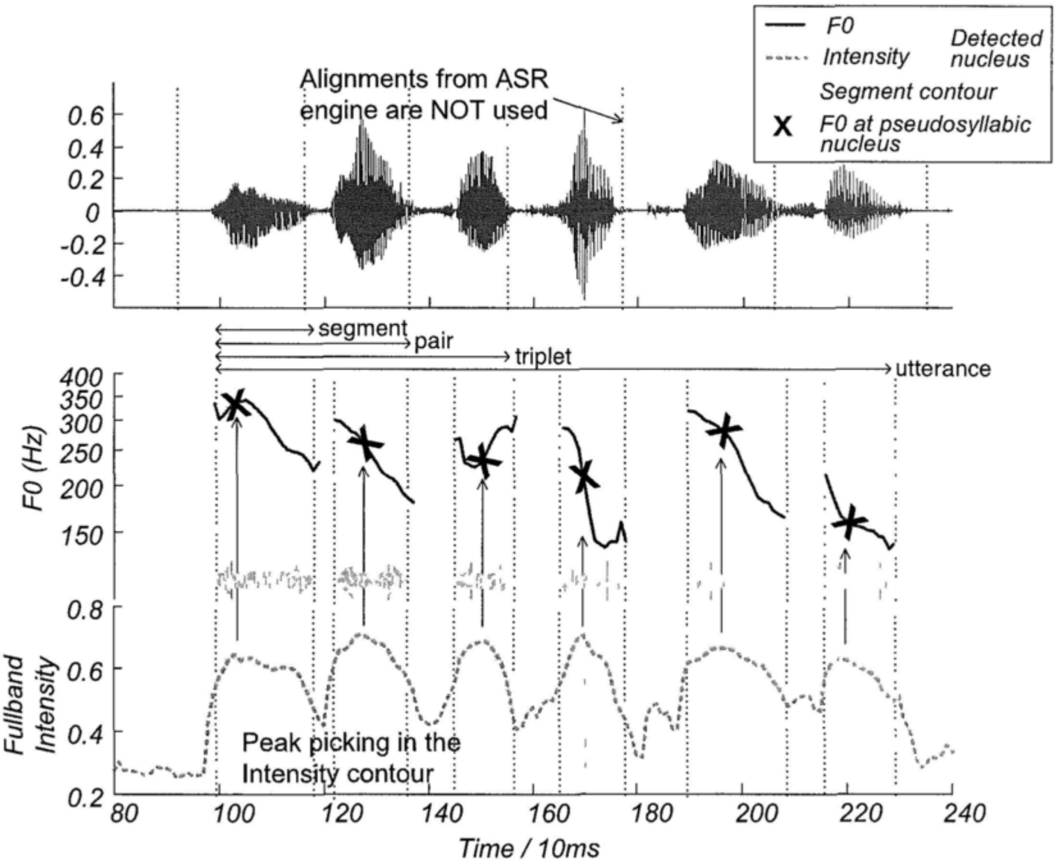


Figure 3.6: *Derivation of F0 and intensity segment contours under PPS*

3.3.2 F0, intensity and duration measurements

Prosodic attributes in group (I) **F0 basic**, (II) **Intensity basic** and (III) **Duration basic** are attributes in basic forms. A number of measurements are extracted from each group. The measurements are classified into *frame-based* and *syllable-based.*

Frame-based measurements

Frame-based measurements are those measurements which are directly taken from the segment contour $\boldsymbol{f}$. For each pseudosyllable, the segment contour gives many (normally, several to dozens of) frame-level values, out of which one measurement is taken as the frame-based measurement.

There are three examples of frame-based measurements for F0 and intensity respectively, namely *F0 nucleus*, *intensity nucleus*, *F0 maximum*, *intensity maximum*, *F0 minimum* and *intensity minimum.* They are explained as in Figure

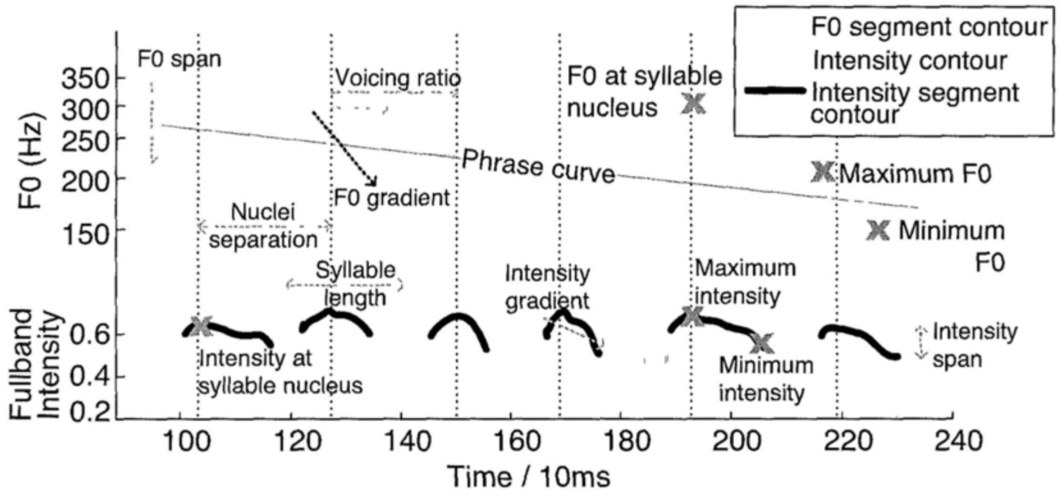


Figure 3.7: *Basic attributes in F0, duration and intensity types*

3.7. *F0/intensity nucleus* is the element in $\boldsymbol{f}$ at the position of pseudosyllabic nucleus. *F0/intensity maximum* and *F0/intensity minimum* are respectively the 95th-percentile and 5th-percentile values in the contour $\boldsymbol{f}$.

Syllable-based measurements

Syllable-based measurements are not directly taken from the segment contour $\boldsymbol{f}$. Normally, they are derived as a function of $\boldsymbol{f}$. There are two syllable-based measurements for F0 and intensity respectively. All duration measurements are syllable-based. They are graphically illustrated in Figure 3.7.

For F0 and intensity, the measurements are *F0 span*, *intensity span*, *F0 gradient* and *intensity gradient*. *Span* measures the numerical range of the elements in $\boldsymbol{f}$. *Gradient* is computed by the quotient of *span* divided by the temporal offset of *maximum* from *minimum*.

For duration, *nuclei separation* is the separation between two consecutive nuclei. *Syllable length* is the length of a pseudosyllable delimited by local minima in the intensity contour. *Voicing ratio* is the ratio of the segment contour length to *syllable length*. Exceptionally long durations due to utterance breaks are excluded by an outlier detection algorithm. In Figure 3.8, the nucleus of the first syllable in the two detected utterances are marked with solid vertical lines, with which an utterance could be clearly identified.

3.3.3 Normalization

The extraction of the three groups of **basic** attributes (Group (I) to Group (III)) is not completed without normalization. Normalization aims at reducing undesirable bias of feature values caused by irrelevant factors like speaker and style variations. In this study, the raw measurements undergo two different normalization methods: *Bias removal* (abbreviated as B) and *z-normalization* (abbreviated as Z). The normalization window covers three temporal ranges, namely *Triplet*, *Utterance* and *File*. In the following, the normalization of frame-based measurements and syllable-based measurements will be described separately.

Normalization of frame-based measurements

First, we consider a frame-based measurement $f_s[t]$, which is the $(t+1)^{\text{th}}$ element in the segment contour $\boldsymbol{f}_s$ from the pseudosyllable s. As a measurement directly taken from the segment contour, $f_s[t]$ could be compared with all other elements in $\boldsymbol{f}_s$ and even in $\boldsymbol{f}_{s\pm\mathrm{W}}$ (the segment contours of neighbouring pseudosyllables). The frame-based normalized attributes for s are obtained by:

$$\text{Raw}: \qquad v_{\text{raw}} = f_s[t], \tag{3.4}$$

$$\text{Bias removal}: \qquad v_{\text{B}} = \overline{f_s[t]}^{\text{B}} = f_s[t] - \mu, \tag{3.5}$$

$$\text{z-normalization}: \qquad v_{\text{Z}} = \overline{f_s[t]}^{\text{Z}} = \frac{f_s[t] - \mu}{\sigma}, \tag{3.6}$$

where μ and σ^2 are the mean and variance estimated from $\{\boldsymbol{f}_{s-\mathrm{W1}}, \ldots, \boldsymbol{f}_{s+\mathrm{W2}}\}$.

$\boldsymbol{f}$ is a segment contour containing several to dozens of measurements. The normalization window covers W1 + W2 + 1 pseudosyllables in the vicinity of the target syllable s. By varying W1 and W2, three time spans are considered:

- *Triplet*, which covers three consecutive syllables, $\boldsymbol{f}_{s-1}$, $\boldsymbol{f}_s$ and $\boldsymbol{f}_{s+1}$.
- *Utterance*, which covers a pause-delimited utterance (Figure 3.8)
- *File*, the longest available content in test data, which may be 10, 30 or 45 seconds depending on the test conditions.

Normalization recovers the syntagmatic properties of attributes, reducing the bias and dynamic range variations over different time spans. For instance, utterance mean and variance are related to intonation. Statistics of longer time span over an utterance may carry certain speaker characteristics.

There are six frame-based measurements, which are F0/intensity nucleus, F0/intensity maximum and F0/intensity minimum. For each measurement, two normalization methods (B,Z) over three time spans (*Triplet, Utterance, File*) are applied to the log-scale value. For the time span *File*, the normalization is also performed for the linear feature value. Together with the *raw* measurement, each frame-based prosodic measurement gives rise to nine normalized attributes.

Normalization of syllable-based measurements

Syllable-based measurements include span, gradient, nuclei separation, syllable length and voicing ratio. In this section, a syllable-based measurement is represented by the function output $g(\boldsymbol{f})$. The normalized syllable-based attributes are obtained by,

$$\text{Raw}: \qquad v_{\text{raw}} = g(\boldsymbol{f}_s), \tag{3.7}$$

$$\text{Bias removal}: \qquad v_{\text{B}} = \overline{g(\boldsymbol{f}_s)}^{\text{B}} = g(\boldsymbol{f}_s) - \mu, \tag{3.8}$$

$$\text{z-normalization}: \qquad v_{\text{Z}} = \overline{g(\boldsymbol{f}_s)}^{\text{Z}} = \frac{g(\boldsymbol{f}_s) - \mu}{\sigma}. \tag{3.9}$$

The normalization operations are similar to those for frame-based measurements. The normalization window is $\{g(\boldsymbol{f}_{s-\text{W1}}), \ldots, g(\boldsymbol{f}_{s+\text{W2}})\}$. Since $g(\boldsymbol{f})$ is a scalar, the points for calculating μ and σ will be much fewer than the case of frame-based measurements. Normalization over *Triplet* is not done because of insufficient data for mean and variance calculations. Four normalization methods (B, Z in *Utterance* and *File*) are applied to the log-scale value of the features. Together with the *raw* measurement, each syllable-based prosodic feature gives rise to five normalized attributes.

3.3.4 Regression and residue features

F0 gradient motivates the use of regression and residue features. Lin and Wang [60] suggested that the second-order coefficient from the polynomial regression of an F0 contour provided language-dependent information. In this thesis, the first- and second-order regression coefficients are calculated from the F0 and the intensity segment contour. Consider a segment contour $\boldsymbol{f} = [f[0], f[1], ..., f[T-1]]^T$. We perform the M^{th}-order regression of $\boldsymbol{f}$ and obtain a set of regression coefficients $\boldsymbol{a}^* = [a_0^* \; a_1^* \; \cdots \; a_M^*]^T$ by,

$$\boldsymbol{a}^* = \underset{\boldsymbol{a}}{\operatorname{argmin}} \sum_{t=0}^{T-1} \left(f[t] - \sum_{m=0}^{M} a_m t^m \right) . \tag{3.10}$$

a_M^*, the highest-order coefficient in $\boldsymbol{a}^*$, is taken as the prosodic attribute from regression analysis. Let v_{reg1} and v_{reg2} denote the regression attributes in first and second order. They are obtained as a_M^* in $\boldsymbol{a}^*$, after performing regression with $M = 1$ and $M = 2$ in Eq.(3.10) respectively.

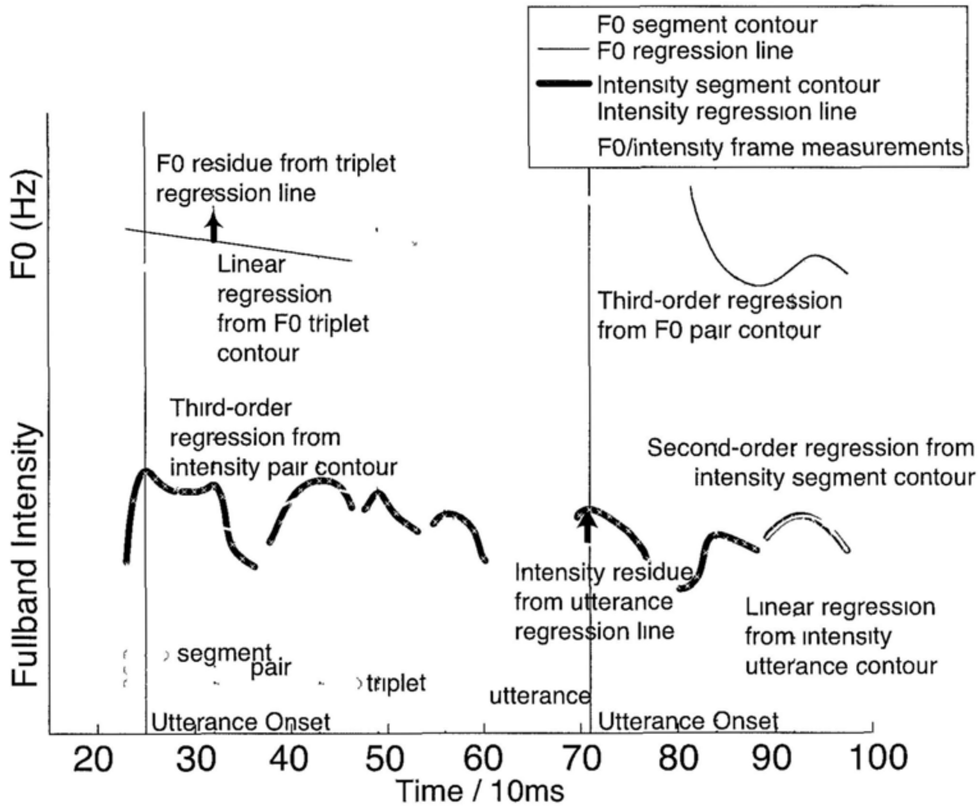


Figure 3.8: *Regression and residue attributes in F0 and intensity types*

Motivated by the supra-tone units for tone modeling [85], regression on *segment contours* is also performed on contours across two syllables (*pair contours*, Fig. 3.8). Up to the fourth-order regression is done to capture the high order of curvature.

Regressions of the contours across three syllables (*Triplet*) and longer regions (*Utterance*) are not intended to model the contour shape. They tend to represent the intonation in a longer temporal range, providing another form of normalization to F0 and intensity. F0 residue and intensity residue are calculated by subtracting the regression line at nucleus from the F0/intensity measurements at the same position, representing syllable-level fluctuations around the phrase curve (Figure 3.8).

3.3.5 Summary of prosodic attributes

To systematically represent the large set of prosodic attributes, a numerical index is assigned to each prosodic attribute to replace the subscript notation. All prosodic attributes used are enumerated in Table 3.2.

Table 3.2: *The unified notation for all prosodic attributes*

	Group Name	Attribute with specified normalization / extraction method		Attribute index (i in v_i)		
				F0-type	Intensity-type	Duration-type
(I,II)	F0/Intensity basic (frame-based)	Nucleus	Raw	31	71	
			Z-File	32	72	
			Z-File (linear)	33	73	
			Z-Utterance	34	74	
			Z-Triplet	35	75	
			B-File	36	76	
			B-File (linear)	37	77	
			B-Utterance	38	78	
			B-Triplet	39	79	
		Maximum	Raw	41	81	
			Z-File	42	82	
			Z-File (linear)	43	83	
			Z-Utterance	44	84	
			Z-Triplet	45	85	
			B-File	46	86	
			B-File (linear)	47	87	
			B-Utterance	48	88	
			B-Triplet	49	89	
		Minimum	Raw	51	91	
			Z-File	52	92	
			Z-File (linear)	53	93	
			Z-Utterance	54	94	
			Z-Triplet	55	95	
			B-File	56	96	
			B-File (linear)	57	97	
			B-Utterance	58	98	
			B-Triplet	59	99	
	F0/Intensity basic (syllable-based)	Span	Raw	61	101	
			Z-File	62	102	
			Z-Utterance	63	103	
			B-File	64	104	
			B-Utterance	65	105	
		Gradient	Raw	66	106	
			Z-File	67	107	
			Z-Utterance	68	108	
			B-File	69	109	
			B-Utterance	70	110	
(III)	Duration basic (syllable-based)	Nuclei Separation	Raw			111
			Z-File			112
			Z-Utterance			113
			B-File			114
			B-Utterance			115
		Syllable length	Raw			116
			Z-File			117
			Z-Utterance			118
			B-File			119
			B-Utterance			120
		Voicing ratio	—			129
(IV,V)	F0/Intensity regression	1st-order on 1 syllable		11	21	
		2nd-order on 1 syllable		12	22	
		1st-order on 2 syllables		16	26	
		2nd-order on 2 syllables		17	27	
		3rd-order on 2 syllables		18	28	
		4th-order on 2 syllables		19	29	
		1st-order on 3 syllables		121	125	
		1st-order on utterance		122	126	
(VI,VII)	F0/Intensity residue	on triplet		123	127	
		on utterance		124	128	

Chapter 4

Statistical language modeling

After the 105 prosodic attributes are extracted, this chapter is devoted to the problem of modeling these attributes. N-gram statistical language modeling is commonly used to capture long-range sequential information. In related studies, pseudosyllable bigram and trigram modeling and simple modeling in the phrase level are typical [12, 46]. In longer ranges, prosodic 4-grams and phonetic 5-grams were used for speaker and dialect recognition respectively [70, 86].

The main idea of this chapter is to construct various n-grams of prosodic attributes. The flow of this chapter follows the modeling process, which is shown in Figure 4.1. Continuous-valued prosodic attributes have low resolutions, it is typical to **quantize** a continuous prosodic feature to discrete categories [12, 56](Section 4.1). These discrete-valued attributes can be used by themselves or combined with other attributes to form **prosodic tokens** (Section 4.2). **Vector space modeling** is adopted to model the statistics of prosodic tokens and their n-grams with count vectors (Section 4.3). With a parallel and flexible approach known as **prosodic attribute model** (PAM, Section 4.4), different count vector constructions will be compared with LID experiments. PAM with vector space modeling gives count vectors with a large dimension. This will bring a huge computation load. **Attribute selection** is carried out to produce a moderately-sized set of attributes. Finally, a standard construction of super term-document matrix is derived from these selected attributes for subsequent LID experiments (Section 4.5).

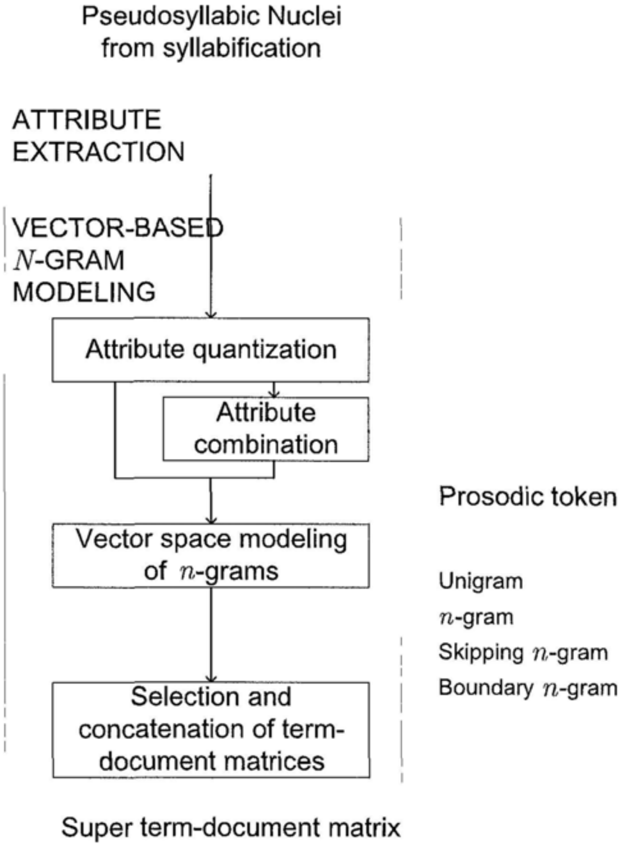

Pseudosyllabic Nuclei from syllabification
ATTRIBUTE EXTRACTION
VECTOR-BASED N-GRAM MODELING
Attribute quantization
Attribute combination
Vector space modeling of n-grams
Selection and concatenation of term-document matrices
Prosodic token
Unigram
n-gram
Skipping n-gram
Boundary n-gram
Super term-document matrix

Figure 4.1: *Statistical language modeling of the tokenized prosodic features*

In this chapter we will encounter language modeling notations and equations for the calculation of term-document matrix dimensions. In short, q refers to a quantized prosodic attribute. The quantization resolution is denoted by $\|\mathcal{Q}\|$, where $\mathcal{Q}$ is the set of all possible values in q. We also introduce *prosodic token*, w. In a loosely defined context, w and q can be perceived similarly as a syllable unit for n-gram modeling. Readers can also refer to the **List of Symbols** pages in the beginning of the thesis.

4.1 Attribute quantization

With the fewest assumptions on the distribution of a feature value, scalar quantization assigns the continuous-valued attribute into equally populated bins [56]. Let $v_{(\imath,s)}$ denote the i^{th} prosodic attribute at the pseudosyllable s. The quantized attribute $q_{(\imath,s)}$ is given by,

$$q_{(i,s)} = q_e\big(v_{(i,s)}\big) \quad \in \mathcal{Q}_i, \tag{4.1}$$

where $q_e(\cdot)$ is the quantization function. The quantization levels can be different for each attribute i. $\mathcal{Q}_i$ is the set of all possible values of the quantized attribute. It would be referred to as *inventory* hereinafter. The *cardinality* of this set, denoted by $\|\mathcal{Q}_i\|$, is equivalent to the *size of the inventory* or the quantization resolution.

The use of equally-populated bins implies that the quantization is not uniform. In this study, the decision levels are found from a multilingual data set comprising 548k pseudosyllables from the NIST LRE 1996 development and evaluation sets, NIST LRE 2003 evaluation set and OGI-TS corpus [87, 88, 89]. These multi-lingual data sets include speech data from English, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Russian, Spanish, Tamil and Vietnamese.

A prosodic LID system previously reported used only two to three bins for quantizing each prosodic feature [12]. On the other hand, a study using prosodic features for speaker recognition used up to 60 quantization bins. Significant performance gain was reported when the number of quantization levels were increased from 2 to 5 [56]. In this thesis, three quantization resolutions, 3, 6 and 9, are tried.

4.2 Generation of prosodic tokens

A set of *prosodic tokens* can be defined by one or more quantized attributes. Let w_s denote the prosodic token for pseudosyllable indexed s. For clarity in the following discussions about n-gram modeling, the defining attributes, i, will be omitted in the notation of the prosodic token.

Prosodic tokens defined by one attribute

This is the simplest case when only a single prosodic attribute is considered. The prosodic token w_s is equal to the quantized prosodic attribute.

$$w_s = q_{(i,s)} \in \mathcal{Q}_i. \tag{4.2}$$

The physical meaning of a *prosodic token* can be understood by looking at an example in which F0 gradient is the defining attribute. Such a token set includes some "rising tone tokens", some "flat tone tokens" and some "falling tone tokens".

Prosodic tokens defined by multiple attributes

In some cases it is beneficial to model two or more attributes together. The set of prosodic tokens are formed by taking the Cartesian product among the individual attributes concerned. Taking the co-modeling of two attributes as an example, we have

$$w_s = q_{(i,s)} \times q_{(j,s)} \in \mathcal{Q}_i \times \mathcal{Q}_j, \tag{4.3}$$

where i and j are the indices of two different prosodic attributes. For example, we can look at the high-rising and low-rising tones in Cantonese. Their characteristics are described by the F0 attribute together with the F0 gradient attribute. By defining a set of *F0 register-gradient* tokens, the properties of the mentioned tones can be explicitly modeled. Other examples of multiple-attribute prosodic tokens include taking the Cartesian product of two attributes with the same measurement but different normalization methods.

4.3 Vector space modeling of prosodic token n-grams

A *bag of sounds* approach is adopted from previous works on information retrieval, particularly text categorization [4, 90]. Given many text documents, the task of text categorization is to look for a certain number of characteristic words (or terms) that can classify the documents into different categories. It is a two-mode analysis problem. In the training process, classes in both dimensions, namely *terms* and *documents*, are derived simultaneously [91, 92, 93].

In a prosody-based LID problem, training/testing utterances are analogous to text documents. Prosodic tokens are analogous to words or terms. The use of a *term-document matrix* is adopted to model variable-length speech utterances with fixed-length feature vectors [4, 39, 40, 56, 94]. For each training utterance (document), a *count vector* is constructed. Its elements are the *occurrence counts* of prosodic tokens (terms) in the utterance. The count vectors of different documents are aligned and stacked together to form the term-document matrix.

Figure 4.2 shows two P-by-K term-document matrices computed from the data of two languages n_1 and n_2. The occurrence counts of P prosodic tokens in K documents of each language are modeled. By comparing the column vectors in the two matrices, it is found that the second token from the top has sparse occurrences in language n_1, while the fourth to sixth prosodic tokens in language n_2 have either frequent or sparse occurrences. These are language-specific properties useful for LID. Some terms, such as the third one from the top, have similar occurrence patterns in both languages. They can be discarded without affecting recognition performance much.

In practical implementation, the occurrence counts of prosodic token n-grams are also taken into accounts to model long-range sequential information [40, 95, 96]. The number of terms (P) in the term-document matrix depends on the size of the prosodic token inventory, and the order of n-gram. Usually it is much larger than that shown in Figure 4.2. Language classifier can be built with a vector-based classifier, typically support vector machine.

Let $x(:, k)$ denote a count vector for training document k, which is an utter-

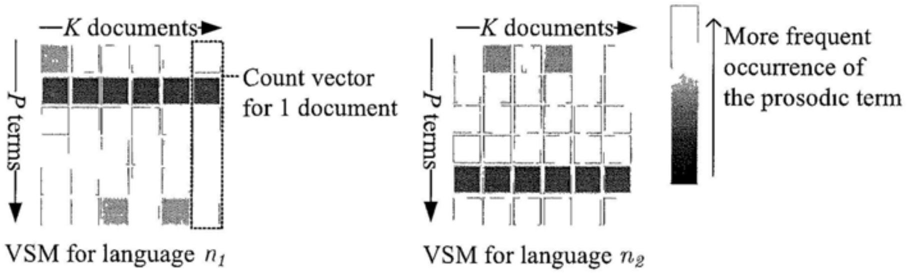


Figure 4.2: *Term-document matrices for utterances spoken in two languages* n_1 *and* n_2

ance, in the term-document matrix. Different ways of constructing the count vectors will be introduced below. Construction of count vectors uses prosodic tokens. In most cases, only one or two prosodic attributes are involved. Theoretically the count vector construction can be repeated with each of the 105 attributes defined in Table 3.2, giving a large number of term-document matrices.

4.3.1 Count vectors for different prosodic token n-grams

Prosodic token unigrams

The count vector of a prosodic token unigram is constructed by counting the occurrence of each prosodic token. The dimension of the count vector is equal to $\|\mathcal{Q}_i\|$. For a prosodic token defined by an attribute pair i and j, there are $\|\mathcal{Q}_i \times \mathcal{Q}_j\|$ different tokens to be counted.

If a prosodic token w_s is defined by single attribute, its count in the utterance (document) k is given by,

$$x(w_s, k) = \frac{C(w_s|k)}{\sum_{w_t \in \mathcal{Q}_i} C(w_t|k)}. \tag{4.4}$$

If w_s is defined by an attribute pair, the count is given by,

$$x(w_s, k) = \frac{C(w_s|k)}{\sum_{w_t \in \mathcal{Q}_i \times \mathcal{Q}_j} C(w_t|k)}. \tag{4.5}$$

$C(w_t|k)$ is a function that returns the actual count. It is normalized by the sum of counts of all unique tokens w_t in the prosodic token inventory ($\mathcal{Q}_i$ or $\mathcal{Q}_i \times \mathcal{Q}_j$).

Prosodic token n-grams

Long-range sequential information can be modeled by prosodic token n-grams. n-gram modeling of different attributes are done separately. Assume the *inventory size* of attribute i is 6 (i.e. $\|\mathcal{Q}_i\| = 6$). If the order of n-gram is 3, prosodic token trigram of this attribute has the *inventory size* of $\|{\mathcal{Q}_i}^3\| = 216$. The count for a prosodic token n-gram $\hat{w}_s w_s$ in document k is obtained as,

$$x(\hat{w}_s w_s, k) = \frac{C(\hat{w}_s w_s|k)}{\sum_{\hat{w}_t w_t \in \mathcal{Q}_\imath{}^n} C(\hat{w}_t w_t|k)}, \tag{4.6}$$

where $\hat{w}_s = w_{s-(n-1)}, \ldots, w_{s-1}$ is the history of $n-1$ preceding pseudosyllables.

Skipping n-grams

In the modeling of higher-order n-grams, data scarcity is a concern. It is desirable to avoid zero probability estimates, which happen when an n-gram does not occur in the training data [96]. Assume we have $w_{s-2}w_{s-1}w_s$ representing the trigram of a prosodic token. The trigram count is given by,

$$x(w_{s-2}w_{s-1}w_s, k) = \frac{C(w_{s-2}w_{s-1}w_s|k)}{\sum_{w_{t-2}w_{t-1}w_t \in \mathcal{Q}_\imath{}^3} C(w_{t-2}w_{t-1}w_t|k)}. \tag{4.7}$$

When the order of n-gram increases, the prosodic token inventory $\mathcal{Q}_\imath{}^n$ increases exponentially. The dynamic range of x becomes huge and the data scarcity may sabotage the representative power of the counts.

Various measures are taken to maintain the robustness of the counts obtained. The most well-known technique is smoothing. It takes some probability away in the rarely occurred n-grams for which the probability is greatly overestimated. Meanwhile, additive smoothing finds a constant, which is added to all unigram counts and thus eliminate zero probability items [96]. Moreover, a squash function can be used to normalize the dynamic range of the terms, and to ensure that the n-grams with large probabilities would not dominate [40]. In document retrieval studies, the counts of word n-grams are scaled by *inverse-document frequencies* [39]. In another approach, models combining classes and words can be used, resulting fewer units and relieving the pressure towards scarcity [97, 98].

Skipping n-gram modeling is a rather simple approach to tackling data scarcity. In this approach, an n-gram is broken into subsets of lower-order n-grams by skipping some of the tokens, so as to reduce zero-probability occurrences. Examples of using this method for prosodic features are found in [45], where a prosodic token trigram was broken into three skipping trigrams in the order of 2,

$$w_{s-2}w_{s-1}w_s \longrightarrow w_{s-2}w_{s-1}, w_{s-2}w_s, w_{s-1}w_s. \tag{4.8}$$

In this thesis, a count for a prosodic token n-gram will be broken into C_2^n counts in its skipping n-grams. Take the prosodic token trigram $w_{s-2}w_{s-1}w_s$ as an example again, the skipping n-gram prosodic token counts are computed as,

$$x(w_{s-2}w_{s-1}, k) = \frac{C(w_{s-2}w_{s-1}|k)}{\sum_{w_{t-2}w_{t-1} \in \mathcal{Q}_\iota{}^2} C(w_{t-2}w_{t-1}|k)}, \tag{4.9}$$

$$x(w_{s-2}w_s, k) = \frac{C(w_{s-2}w_s|k)}{\sum_{w_{t-2}w_t \in \mathcal{Q}_\iota{}^2} C(w_{t-2}w_t|k)}, \tag{4.10}$$

$$x(w_{s-1}w_s, k) = \frac{C(w_{s-1}w_s|k)}{\sum_{w_{t-1}w_t \in \mathcal{Q}_\iota{}^2} C(w_{t-1}w_t|k)}. \tag{4.11}$$

The skipping n-gram approach not only alleviates the threat of data scarcity, but also reduces the dimension of the count vector. In regular n-gram modeling, the inventory size is $\|\mathcal{Q}_\iota{}^n\|$. With C_2^n sets of skipping n-grams in the order of 2, the inventory size is $C_2^n \times \|\mathcal{Q}_\iota{}^2\|$. Dimension reduction is more noticeable when $\|\mathcal{Q}_\iota\|$ is large and/or the order of n-gram is high.

Boundary n-grams

Boundary refers to a sentence boundary. In long-range modeling, it is more likely that a high-order n-gram touches or spans across a sentence boundary. There were very few related studies on prosody modeling that considered boundaries. In [99], consistent reductions of word recognition errors were demonstrated by incorporating sentence boundary information into prosody modeling. For language recognition, pragmatic functions like speaker's intention and attitude are often expressed near sentence boundaries and may create noise. Examples include tone patterns for interrogations and exclamations. Meanwhile, if language-specific boundary tones exist, modeling sentence boundary would benefit language recognition.

Consider a *boundary bigram*, which is defined as a bigram at sentence initial $(\#_{s-1}w_s)$ or at sentence final $(w_{s-1}\#_s)$, where "#" denotes a sentence boundary.

It is inferred from detected short pauses in speech. By accommodating the skipping model concept, *boundary skipping n-grams* can be defined, which are skipping n-grams in the order of 2 (e.g. $\#_{s-m}w_s$ or $w_{s-m}\#_s$ where $m < n$). Three configurations with different treatments to *boundary bigrams* are tested:

[**IGNORE**] Ignore boundary: This is the simplest approach. Automatic pause detection is not carried out and no boundary bigram is computed.

[**DELETE**] Delete boundary bigrams: The boundary bigrams ($\#_{s-m}w_s$ or $w_{s-m}\#_s$) are dropped before any statistical modeling is done. This is based on the assumption that boundary bigrams mainly carry pragmatic functions unrelated to languages.

[**EXPLOIT**] Exploit boundary bigrams: A separate prosodic token of pause is used in the modeling of boundary bigrams. For instance, with the *inventory size* being equal to 6, there are $(6 + 1)^2 = 49$ bigrams, among which 36 are normal bigrams, 6 are sentence initial bigrams, 6 are sentence final bigrams, and 1 is a pause bigram ($\#_{s-m}\#_s$).

4.3.2 Super term-document matrix

One important assumption in the generation of prosodic token is to process each attribute separately by scalar quantization [12, 56]. Thus, many *term-document matrices* are constructed in parallel. Each of them models only one to several attributes. This results numerous term-document matrices and they are concatenated to form a *super term-document matrix*. Figure 4.3 illustrates a *super term-document matrix* whose component matrices include two matrices constructed by prosodic token unigrams and one matrix constructed by prosodic token skipping trigrams (Section 4.3.1).

4.4 Prosodic attribute model

Vector space modeling of prosodic attributes implemented in this thesis can process a large number of prosodic attributes. In the construction of super term-document matrices, prosodic token unigram and prosodic token skipping

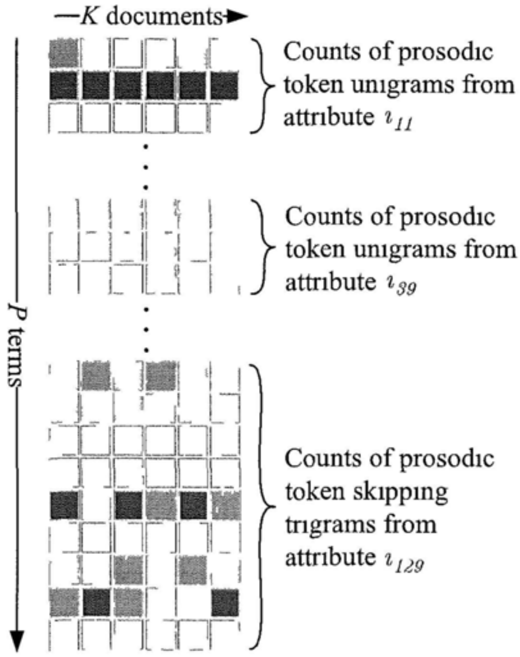

—K documents→
Counts of prosodic token unigrams from attribute i_{11}
Counts of prosodic token unigrams from attribute i_{39}
Counts of prosodic token skipping trigrams from attribute i_{129}
P terms

Figure 4.3: *Concatenation to form a super term-document matrix, subscripts to i are attribute indices in Table 3.2*

trigram can be selected in a flexible manner. Also, modeling of different prosodic attributes are done in a parallel manner. This flexible and parallel modeling approach is referred to as *prosodic attribute model* (PAM) hereinafter.

In this section, we will explore a preliminary design of term-document matrix under PAM. The parallel modeling of prosodic features is common to the studies of prosodic features [12, 56]. We will use an experiment to show how the modeling capabilities suffer by separate modeling instead of modeling prosodic features altogether. We also want to construct compact term-document matrices while keeping the quantization resolution and the order of n-gram high enough to retain necessary information. Finally, we want to compare the simple prosodic tokens which are defined by a single attribute (Section 4.2) with the complex ones defined by several attributes (Section 4.2). Constraining a fixed matrix dimension, we want to find out which combination strategy is more effective.

In the following experiments, different ways of constructing term-document matrices will be compared in terms of LID performance in NIST Language Recognition Evaluation (LRE) 2009 [14]. Comparison will be based on the *average equal error rate* (*average EER*) Term-document matrix constructions which give smaller *average EER* are preferred The derivation of *average EER* will not be discussed for the time being, as the lack of these details does not obscure the major focus of comparing different term-document matrix constructions

4.4.1 Attribute-wise modeling in PAM

PAM adopts the vector space modeling techniques from a phonotactic LID system [4, 39] Modeling of prosodic attributes differs from the phonotactic counterpart in that a phonetic token is normally the output of a phone recognizer, and such a token is *completely defined by different phonetic dimensions* For instance, the Cantonese phone $/\mathrm{ts}^{\mathrm{h}}/$ in Section 3 1 is defined not only by its *manner of articulation* (affricate), but also by its *place of articulation* (alveolar, articulated with the tongue against or close to the superior alveolar ridge), as well as by other dimensions (voiceless, aspirated) On the contrary, a prosodic token is *partially defined*, in the sense that its acoustic correlates lie on only a subset of, but not all prosodic attributes

In this section, experimental results will show how the modeling capabilities suffer from the attribute-by-attribute separate modeling of PAM We will create *prosodic phones* to compare with the prosodic tokens in PAM *Prosodic phones* are completely defined on all prosodic attributes, analogous to how the phone $/\mathrm{ts}^{\mathrm{h}}/$ is defined in the previous paragraph In the prosodic domain, there is no standard way to come up with such a phone Intuitively, it is derived by taking the Cartesian product of all defining attributes (same method as in Section 4 2) To work with a manageable dimension, we assume a concise attribute set to represent the whole prosodic space, and modeling is done up to bigram The attributes in the concise set are listed in Table 4 1

These three attributes are considered to be representative for the three major types of prosodic attributes, namely *F0*, *Intensity* and *Duration* In a later section (Section 4 5 2), attribute analysis will be performed to show that these attributes are among the most effective attributes for LID

Table 4.1: *The defining attributes of a prosodic phone*

$i^\sharp$	Type	Attribute
39	F0	*Nucleus*, normalized with *Bias removal* over *Triplet*
27	Intensity	2^{nd}-order regression on 2 pseudosyllables
129	Duration	Voicing ratio

$^\sharp i$ is the attribute index in Table 3.2

In unigram modeling, the number of *terms* in the term-document matrix is equal to the size of the *prosodic phone* inventory. For bigram modeling, the inventory size is squared. If we model *prosodic phones*, the total dimension of the term-document matrix is,

$$\prod_{i=[39,27,129]} \|\mathcal{Q}_{(i,\text{1-gram})}\| + \left(\prod_{i=[39,27,129]} \|\mathcal{Q}_{(i,\text{2-gram})}\| \right)^2 . \qquad (4.12)$$

$\|\mathcal{Q}_{(i,\text{1-gram})}\|$ and $\|\mathcal{Q}_{(i,\text{2-gram})}\|$ are the scalar quantization resolutions for the defining attributes in unigram and bigram construction respectively. The two resolutions can be different. For all attributes, $\|\mathcal{Q}_{(i,\text{1-gram})}\|$ is set to 9 to secure a higher resolution. $\|\mathcal{Q}_{(i,\text{2-gram})}\|$ is set to 3 so that the number of bigrams is at a manageable size. The total dimension is 1458.

With PAM, the term-document matrices for the three attributes are constructed separately. The super term-document matrix is obtained by concatenating the attribute-wise term-document matrices. Its dimension is,

$$\sum_{i=[39,27,129]} (\|\mathcal{Q}_{(i,\text{1-gram})}\| + \|\mathcal{Q}_{(i,\text{2-gram})}\|^2). \qquad (4.13)$$

With $\|\mathcal{Q}_{(i,\text{1-gram})}\| = 9$ and $\|\mathcal{Q}_{(i,\text{2-gram})}\| = 3$, the total dimension with PAM is only 54.

Table 4.2 compares the approach using completely defined *prosodic phones* against the approach with partially defined *prosodic tokens* with PAM. Comparison is made in terms of the number of terms in the super term-document matrices, as well as the average EER. To the best of our knowledge, there has not been any similar comparison. For both *prosodic phones* and *prosodic tokens*

Table 4.2: *EER with different models for NIST LRE 2009*

Modeling method	Number of terms♯	Average EER
Prosodic phones	1458*	32.38%
Prosodic tokens with PAM	54*	34.76%

* $\|\mathcal{Q}_{(i,1\text{-gram})}\| = 9$ and $\|\mathcal{Q}_{(i,2\text{-gram})}\| = 3$

♯ Number of terms is the dimension of the super term-document matrix

with PAM, the average EER is 30%-35%. Although we only use a concise attribute set with 3 prosodic attributes, this result is already comparable to those reported in other studies [12, 46]. PAM achieves significant dimension reduction by discarding the across-attribute information. With three attributes, it can be shown that PAM reduces the term-document matrix dimension by 96%. Given the compactness of PAM, we see a great potential to extend its modeling capabilities by including other prosodic attributes. The attribute-wise modeling in PAM is justified and will be applied to other attributes.

4.4.2 Expanding single-attribute prosodic tokens

Based on the attribute-wise separate modeling in PAM (Eq.(4.13)), there are a couple of options to improve the representation of information in the term-document matrices. We can increase the quantization resolution, or increase the order of n-gram to model trigram information. Consider two cases,

$$\sum_{i=[39,27,129]} \left(\|\mathcal{Q}_{(i,1\text{-gram})}\| + \|\mathcal{Q}_{(i,n\text{-gram})}\|^n\right)$$

$$\text{Increased resolution: } n = 2, \|\mathcal{Q}_{(i,n\text{-gram})}\| = 6.$$

$$\text{Trigram modeling: } \quad n = 3, \|\mathcal{Q}_{(i,n\text{-gram})}\| = 3.$$

The dimensions of the super term-document matrices for the above two cases (Increased resolution, Trigram modeling) are 135 and 108 respectively. The average EER in Table 4.3 show that the term-document matrix with *increased resolutions* gives a slightly better performance than that with longer-range information. While this trend is general to most languages, in our study the

Table 4.3: *EER with increased resolution vs trigram modeling (NIST LRE 2009)*

Expanded model	Number of terms♯	Average EER
Increased resolution	135	33.87%
Trigram modeling	108	34.34%

♯ Number of terms is the dimension of the super term-document matrix

detections of Hindi, Portuguese, Spanish and Turkish are found to benefit more from trigram modeling. In the following experiments, prosodic tokens are modeled up to their trigram. The quantization resolution will be set to 6.

4.4.3 Combining single-attribute prosodic tokens

In Section 4.2, multiple attributes are combined to form prosodic token unigrams. In Section 4.3.1, prosodic token n-grams are constructed by taking the Cartesian products of neighbouring unigrams. We would like to compare whether one combination method is favourable over another.

With the concise attribute set (Table 4.1), the size of super term-document matrix for the two combination approaches is determined as follows,

Attribute combination — Unigram + multiple-attribute unigram (in pseudosyllabic positions $s-1$, s and $s+1$),

$$\sum_{i=[39,27,129]} \|\mathcal{Q}_{(i,\text{1-gram})}\| + 3 \times \prod_{i=[39,27,129]} \|\mathcal{Q}_{(i,\text{1-gram})}\|. \tag{4.14}$$

n-gram construction — Unigram + single-attribute prosodic token trigrams,

$$\sum_{i=[39,27,129]} \|\mathcal{Q}_{(i,\text{1-gram})}\| + \sum_{i=[39,27,129]} \|\mathcal{Q}_{(i,\text{3-gram})}\|^3. \tag{4.15}$$

In both Eq.(4.14) and (4.15), the first term corresponds to the prosodic token unigram. In the second term, nine parameters (three attributes in three pseudosyllabic positions) are combined in two different ways. By retaining the first term on prosodic token unigram and rearranging the second term, we also tried two *irregular combinations* of the nine parameters. Following the notation on prosodic tokens (Eq.(4.2) and (4.3)), in the irregular combinations we have the following prosodic tokens,

Irregular combination 1,

$$\begin{gathered} q_{(39,s)}, q_{(27,s)}, q_{(129,s)}, \\ q_{(39,s)} \times q_{(27,s+1)} \times q_{(129,s-1)}, \\ q_{(39,s-1)} \times q_{(27,s)} \times q_{(129,s+1)}, \\ q_{(39,s+1)} \times q_{(27,s-1)} \times q_{(129,s)}. \end{gathered} \tag{4.16}$$

Irregular combination 2,

$$\begin{gathered} q_{(39,s)}, q_{(27,s)}, q_{(129,s)}, \\ q_{(39,s)} \times q_{(27,s-1)} \times q_{(129,s+1)}, \\ q_{(39,s+1)} \times q_{(27,s)} \times q_{(129,s-1)}, \\ q_{(39,s-1)} \times q_{(27,s+1)} \times q_{(129,s)}. \end{gathered} \tag{4.17}$$

Note that in all combinations, the same attributes are used and hence the dimension of the super term-document matrices is the same. The four combinations can be interpreted as modeling the joint statistics of different factors. Intuitively, if two factors are independent, joint statistics modeling is not necessary. Among the four combination methods, the *n-gram construction* method is expected to give the best result. This is because a high correlation is expected within the n-gram of a single attribute.

Table 4.4 shows the LID performances of the four combinations. The results agree with our intuition. *N-gram construction* gives slightly lower average EER than other combination methods. It should be noted that this construction also attains a lower average EER compared with the *prosodic phone* modeling method in Table 4.2, with the number of terms in term-document matrices reduced by half.

Table 4.4: *EER with different combinations of prosodic tokens (NIST LRE 2009)*

Combination method	Number of terms♯	Average EER
Attribute combination	675	33.15%
n-gram construction	675	32.34%
Irregular combination 1	675	33.17%
Irregular combination 2	675	33.88%

♯ Number of terms is the dimension of the super term-document matrix

4.4.4 A preliminary design of term-document matrix under PAM

From the above experiments, we conclude the following principles of constructing the term-document matrices under PAM:

- Term-document matrices are created with attribute-by-attribute separate modeling. Matrices from different attributes are concatenated to form a super term-document matrix. (Section 4.4.1)
- Single-attribute prosodic tokens are modeled in *unigram* and *trigram*. The quantization resolutions for the prosodic tokens, $\|\mathcal{Q}_{(i,\text{1-gram})}\|$ and $\|\mathcal{Q}_{(i,\text{3-gram})}\|$ are 9 and 6 respectively. (Section 4.4.2)
- Term-document matrices constructed by prosodic token trigrams are included. These trigrams are combinations of neighbouring unigrams of the same attribute. (Section 4.4.3)

4.5 Attribute selection in term-document matrices

Recall that the construction of term-document matrices is replicated for different prosodic attributes. Among the 105 prosodic attributes introduced in Section 3.3, many of them carry similar kind of information. F0/Intensity/Duration basic attributes exploit various normalization methods, while the regression attributes cover different regression orders on target contours with different lengths. It would be a computation intensive task to exhaust all possible attributes in the super term-document matrix. In this section, an information-theoretic metric is developed to evaluate the prosodic token unigrams of all attributes. From the 105 prosodic attributes (seven groups) in Table 3.2, a subset of prosodic attributes which are effective to LID are selected for the LID experiments.

4.5.1 Mutual information evaluation

The process of attribute selection follows a mutual information approach. The robustness of the method lies on the fact that it measures arbitrary dependencies between the analysis variables, such that it can be applied as a frontend process before classifier training. It is also suitable for classification tasks with complex decision boundaries [100].

In each step of the analysis we focus on one target language, n_t. The full data set can then be divided into two partitions: the true part l_{n_t} which belongs to n_t and the imposter part $\neg l_{n_t}$ which does not.

Evaluation to single prosodic attributes

We adopt an information-theoretic perspective and re-interpret the prosodic tokens defined by a single attribute, as well as the language labels. The quantized attribute q_i is regarded as the observation output from a random variable Q_i. l_{n_t} and $\neg l_{n_t}$ are regarded as the observation output of a binary random variable L_{n_t}. As such, we can quantify the information that Q_i contains about L_{n_t}, by considering the entropy of L_{n_t} conditioned on Q_i. This entropy measure is known as *mutual information* [100]. It is formulated mathematically as,

$$I(L_{n_t}; Q_i) = H(L_{n_t}) - H(L_{n_t}|Q_i), \tag{4.18}$$

where $H(L_{n_t})$ and $H(L_{n_t}|Q_i)$ are entropy terms defined as,

$$H(L_{n_t}) = -\sum_{l_{n_t}=\{0,1\}} P(l_{n_t}) \log P(l_{n_t}), \tag{4.19}$$

$$H(L_{n_t}|Q_\imath) = -\sum_{q \in \mathcal{Q}_\imath} P(q) \left(\sum_{l_{n_t}=\{0,1\}} P(l_{n_t}|q) \log P(l_{n_t}|q) \right). \tag{4.20}$$

Simply speaking, the mutual information equation visits every attribute value q and compares the probability of the true class data $P(l_{n_t})$ and the imposter class data $P(\neg l_{n_t})$. $\mathcal{Q}_\imath$ is the set of all possible values of q (i.e. the inventory). In prosodic token unigram analysis, the size of $\mathcal{Q}_\imath$ is essentially the quantization resolution of attribute i. In this thesis, three resolutions (3, 6 and 9) are investigated and three mutual information values are obtained by Eq.(4.18). The three values are averaged to yield a single mutual information metric for comparison.

The larger the mutual information $I(L_{n_t}; Q_\imath)$, the more information about L_{n_t} is available in $Q_\imath$. For different $Q_\imath$, $I(L_{n_t}; Q_\imath)$ have different order of magnitudes and dynamic ranges. In order to alleviate this problem, $I(L_{n_t}; Q_\imath)$ is compared with $I(R^*; Q_\imath)$ where R^* is a binary random variable independent of $Q_\imath$, but shares similar statistical properties with L_{n_t}. Nevertheless, there does not exist such a binary random variable R^*. In implementation, we create 500 binary random variables $R_1, R_2, ..., R_{500}$. Every R has the same first-order statistics with L_{n_t}, but the mutual information between an attribute $Q_\imath$ and different binary random variable R varies. From the 500 mutual information terms, the first- and second-order statistics are calculated. These statistics are used as normalization references for $I(L_{n_t}; Q_\imath)$. z values for $I(L_{n_t}; Q_\imath)$ with respect to $E_R[I(R; Q_\imath)]$ and $STD_R[I(R; Q_\imath)]$ are calculated as,

$$z_{(n_t,\imath)} = \frac{I(L_{n_t}; Q_\imath) - E_R[I(R; Q_\imath)]}{STD_R[I(R; Q_\imath)]} \tag{4.21}$$

Mutual information $I(\cdot;\cdot)$ is non-negative. A prosodic attribute i should carry more information about L_{n_t} than any random label R. Thus, $z_{(n_t,\imath)}$ is expected to have positive values. A larger value of $z_{(n_t,\imath)}$ indicates attributes i is more effective in identifying language n_t.

Evaluation to prosodic attribute pairs

In Section 4.2, some prosodic tokens defined by multiple attributes are introduced. We consider a prosodic token defined by two attributes i and j. Mutual information to these prosodic tokens can be computed in a similar manner to Eq.(4.18),

$$I(L_{n_t}; Q_{\imath} \times Q_{\jmath}) = H(Q_{\imath}, Q_{\jmath}) - H(Q_{\imath}, Q_{\jmath} | L_{n_t}). \tag{4.22}$$

The inventory is $\mathcal{Q}_{\imath} \times \mathcal{Q}_{\jmath}$. It is the Cartesian product of the inventory of individual attributes i and j. It is assumed that the quantization resolution of both attributes are 6, thus $\|\mathcal{Q}_{\imath} \times \mathcal{Q}_{\jmath}\| = 36$.

The important question is whether the prosodic tokens defined by two attributes would create an effect of synergy. In other words, we want to compare Eq.(4.22) with Eq.(4.18), where $Q_{\imath}$ and $Q_{\jmath}$ are modeled separately. Define $Isyn_{(n_t,\imath,\jmath)}$ as a measure for the degree of synergy between attributes i and $\jmath$ in representing n_t,

$$\begin{aligned}
Isyn_{(n_t,\imath,\jmath)} =& I(L_{n_t}; Q_{\imath} \times Q_{\jmath}) - I(L_{n_t}; Q_{\imath}) - I(L_{n_t}; Q_{\jmath}) \\
=& H(Q_{\imath}, Q_{\jmath}) - H(Q_{\imath}, Q_{\jmath} | L_{n_t}) - (H(Q_{\imath}) - H(Q_{\imath} | L_{n_t})) - (H(Q_{\jmath}) - H(Q_{\jmath} | L_{n_t})) \\
=& (H(Q_{\imath} | L_{n_t}) + H(Q_{\jmath} | L_{n_t}) - H(Q_{\imath}, Q_{\jmath} | L_{n_t})) - (H(Q_{\imath}) + H(Q_{\imath}) - H(Q_{\imath}, Q_{\jmath})) \\
=& I(Q_{\imath}; Q_{\jmath} | L_{n_t}) - I(Q_{\imath}; Q_{\jmath}).
\end{aligned} \tag{4.23}$$

Repeating Eq.(4.23) in all pairs from the 105 attributes, there are $C_2^{105} = 5565$ $Isyn$ metrics, indicating the effectiveness of the 5565 pairs of attributes to the identification of n_t. When $Isyn_{(n_t,\imath,\jmath)}$ gives a large value, it means that it is advantageous to combine attributes i and j to form a prosodic token. In order to attain such a condition, conditional mutual information between $Q_{\imath}$ and $Q_{\jmath}$ should be large while the general mutual information should be small. It means that within the class $l_{n_t} = 1$ or $l_{n_t} = 0$, two attributes i and j should be sharing some information. This information is language dependent, because the mutual information term $I(Q_{\imath}; Q_{\jmath} | L_{n_t})$ is conditioned on the class l_{n_t}. Meanwhile, in the global sense $Q_{\imath}$ and $Q_{\jmath}$ should be independent to the greatest extent, so complementary information among the attributes exists.

4.5.2 Language-independent prosodic attribute selection

Recall there are 105 prosodic attributes (Table 3.2). They fall into seven groups. In single attribute analysis, attribute selection will be done within each group. We select among attributes with different normalization methods from Group (I) to Group (III) attributes. From Group (IV) to Group (VII), attribute selection is done among different regression orders and different temporal ranges in modeling. Attribute selection is done in a language-independent manner, the $z_{(n_t,\imath)}$ metric for different languages n_t are averaged to give $\overline{z_{(n_t,\imath)}}$. The attributes with the large values of $\overline{z_{(n_t,\imath)}}$ within each group will be selected.

For the attribute pair analysis, the *Isyn* metric averaged over different target languages is evaluated for the 5565 attribute pairs. The pairs are ranked in descending orders of the metric. To maintain a manageable size of super term-document matrix, no more than 20 attribute pairs will be selected.

Mutual information metrics are calculated from the training corpora, which are NIST LRE 1996 development and evaluation sets (30-second). The corpora contain data of 12 target languages, including Arabic, English, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Tamil, Vietnamese and Spanish. In each language, there are 130 utterances (roughly 14400 pseudosyllables) for analysis.

Single prosodic attributes

From the 105 prosodic attributes, 20 prosodic attributes that are most effective to LID are selected for the LID experiments. Table 4.5 shows the $\overline{z_{(n_t,\imath)}}$ metric for all prosodic attributes. The selected attributes are marked by check marks.

There are three frame-based measurements in the types of F0 and intensity respectively, namely *nucleus*, *maximum* and *minimum*. *Bias removal* over *Triplets* consistently gives the largest z values and their unigram attributes are selected. These attributes are also selected for prosodic token n-gram modeling. Due to the similarity between these attributes, only the *nucleus* attribute will

Table 4.5: $\overline{z_{(n_t,i)}}$ *scores for different normalization and extraction methods*

Group Name	Attribute with specified normalization / extraction method		F0-type				Intensity-type				Duration-type			
			i	$\overline{z_{(n_t,i)}}$	Selected unigram	Selected n-gram	i	$\overline{z_{(n_t,i)}}$	Selected unigram	Selected n-gram	i	$\overline{z_{(n_t,i)}}$	Selected unigram	Selected n-gram
(I,II)														
F0/Intensity basic (frame based)	Nucleus	Raw	31	2 68			71	2 84						
		Z-File	32	3 78			72	4 41						
		Z-File(linear)	33	4 25			73	3 20						
		Z-Utterance	34	4 18			74	4 38						
		Z-Triplet	35	4 78			75	4 21						
		B-File	36	4 01			76	3 85						
		B-File(linear)	37	4 01			77	2 85						
		B-Utterance	38	4 23			78	4 37						
		B-Triplet	39	5 10	✓	✓	79	4 74	✓	✓				
	Maximum	Raw	41	2 78			81	2 81						
		Z-File	42	3 98			82	4 17						
		Z File(linear)	43	4 22			83	3 09						
		Z-Utterance	44	4 10			84	4 21						
		Z-Triplet	45	4 82			85	3 97						
		B File	46	4 26			86	3 74						
		B-File(linear)	47	4 31			87	2 79						
		B-Utterance	48	4 41			88	4 28						
		B Triplet	49	5 02	✓		89	4 64	✓					
	Minimum	Raw	51	2 78			91	2 84						
		Z-File	52	3 81			92	3 89						
		Z-File(linear)	53	4 74			93	3 78						
		Z-Utterance	54	4 50			94	3 98						
		Z Triplet	55	4 97			95	4 19						
		B-File	56	4 04			96	3 62						
		B-File(linear)	57	4 23			97	3 16						
		B-Utterance	58	4 76			98	3 58						
		B-Triplet	59	5 38	✓		99	3 95	✓					
F0/Intensity basic (syllable-based)	Span	Raw	61	3 19			101	2 73						
		Z File	62	3 28			102	1 31						
		Z-Utterance	63	1 87			103	0 95						
		B-File	64	4 04	✓		104	2 10						
		B-Utterance	65	3 86			105	2 15						
	Gradient	Raw	66	4 35	✓	✓	106	3 03	✓	✓				
		Z-File	67	3 36			107	1 66						
		Z Utterance	68	1 97			108	1 27						
		B-File	69	3 87			109	1 95						
		B-Utterance	70	3 93			110	1 89						
(III)														
Duration basic (syllable-based)	Nuclei Separation	Raw									111	4 34		
		Z-File									112	2 39		
		Z-Utterance									113	1 14		
		B-File									114	3 34	✓	✓
		B-Utterance									115	2 94		
	Syllable length	Raw									116	4 35		
		Z-File									117	1 66		
		Z-Utterance									118	1 78		
		B-File									119	3 08	✓	
		B-Utterance									120	2 78		
	Voicing ratio	—									129	3 96	✓	✓
(IV,V)														
F0/Intensity regression	1^{st}-order on 1 syllable		11	4 82	✓	✓	21	3 26						
	2^{nd}-order on 1 syllable		12	4 94	✓	✓	22	4 39	✓	✓				
	1^{st}-order on 2 syllables		16	4 96	✓	✓	26	4 21	✓	✓				
	2^{nd}-order on 2 syllables		17	4 12			27	4 92	✓	✓				
	3^{rd}-order on 2 syllables		18	4 73			28	4 14						
	4^{th}-order on 2 syllables		19	4 19			29	4 14						
	1^{st}-order on 3 syllables		121	5 29			125	4 04						
	1^{st}-order on utterance		122	4 75			126	3 16						
(VI,VII)														
F0/Intensity residue	on triplet		123	5 01	✓	✓	127	4 37	✓	✓				
	on utterance		124	4 32			128	3 93						

be used in the n-gram form.

For syllable-based measurements, the F0/intensity *gradient* attributes are the most effective without normalization. The optimal normalization method for F0 *span* is *Bias removal* over *Files*, while intensity *span* is not effective with any normalization method. The *span* attributes will be used only in unigram form under the F0 type.

There are three duration attributes. *Voicing ratio* does not require normalization and gives a comparatively large z value among all duration attributes. Across different normalization variants, *nuclei separation* is generally more effective than *syllable length.* The optimal normalization method for both attributes is *Bias removal* over *Files. Syllable length* will be used only in unigram form.

For regression attributes, the three attributes with the largest z values will be selected for the type of F0 and intensity respectively. Generally, F0 regression attributes are more effective with lower regression orders and intensity attributes are more effective with higher regression orders. For residue attributes, residues over *Triplets* are more effective.

Prosodic attribute pairs

Table 4.6 lists the attribute pairs selected for constructing multiple-attribute prosodic tokens. The selection is based on the $Isyn_{(n_t,i,j)}$ metric (Eq.(4.23)). Among the candidates with large values of $Isyn_{(n_t,i,j)}$, some combinations that are physically sound are selected over the others. It is noticed that attribute pairs using the same measurement but different normalization/regression methods tend to give larger values of $Isyn_{(n_t,i,j)}$. Some pairs across two attribute groups are noted for regression attributes.

For the basic frame-based attributes of F0 and Intensity, the three frame-based measurements (*nucleus*, *maximum* and *minimum*) demonstrate the same trend. Attributes with *Z-File* and *B-File* normalization methods are combined. For the basic syllable-based attributes of F0 and intensity, similar attributes normalized over the time span of *File* are combined. For duration attributes, *raw* attributes are combined with normalized attributes. Regression attributes in

Table 4.6: *Selected attribute pairs to combine as prosodic tokens*

Group Name	i Attribute 1	i Attribute 2
(I) F0 basic	33 Nucleus, Z-File(linear)	36 Nucleus, B-File
	43 Maximum, Z-File(linear)	46 Maximum, B-File
	53 Minimum, Z-File(linear)	56 Minimum, B-File
	64 Span, B-File	69 Gradient, B-File
	66 Gradient, Raw	67 Gradient, Z-File
(II) Intensity basic	72 Nucleus, Z-File	77 Nucleus, B-File(linear)
	82 Maximum, Z-File	87 Maximum, B-File(linear)
	104 Span, B-File	109 Gradient, B-File
(III) Duration basic	111 Nuclei separation, Raw	112 Nuclei separation, Z-File
	116 Syllable length, Raw	117 Syllable length, Z-File
	116 Syllable length, Raw	129 Voicing ratio
(IV) F0 regression	11 1st-order on 1 syllable	12 2nd-order on 1 syllable
	11 1st-order on 1 syllable	16 1st-order on 2 syllables
	11* 1st-order on 1 syllable	67* F0 Gradient, Z-File
(V) Intensity regression	21 1st-order on 1 syllable	22 2nd-order on 1 syllable
	22 2nd-order on 1 syllable	28 3rd-order on 2 syllables
(VI) F0 residue	121* 1st-order regression on 3 syllables	123* F0 residue on triplet
(VII) Intensity residue	125* 1st-order regression on 3 syllables	127* Intensity residue on triplet

*Attribute pair across two attribute groups

different regression orders or with different length of the target segment contours are combined. They also combine with residue attributes.

4.5.3 The 14-attribute prosodic feature set

In the following LID experiments, prosodic features will be modeled by a super term-document matrix built by three types of vector space constructions. In the first type, term-document matrices with prosodic token unigrams are constructed with every of the 20 selected single attributes in Table 4.5. The second type includes 14 term-document matrices with prosodic token trigrams from Table 4.5. In the last type, the 18 selected attribute pairs in Table 4.6 will be modeled as prosodic token unigrams.

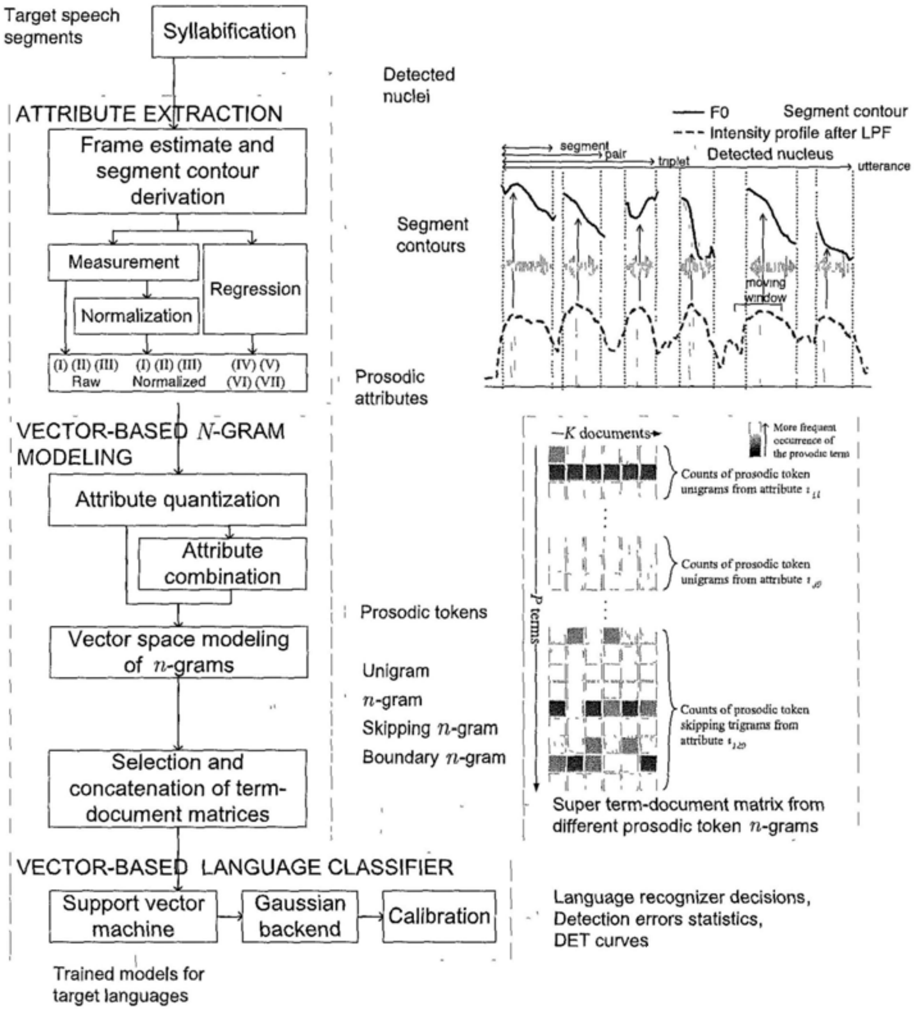
Since there are considerable overlap between the three types of constructions, this super term-document matrix will be named after the number of defining attributes in prosodic token trigrams, i.e. 14-attribute prosodic feature set. Counting the number of terms, the prosodic token trigrams outnumber the other two types and will be the major component in the super term-document matrix.

Chapter 5

Language identification experiments

Before looking at the language identification (LID) experimental results, let us have an overview of the prosody-based LID system as depicted in Figure 5.1. Input speech segments first go through a syllabification process, which locates the nuclei of pseudosyllables (Section 3.1). From these landmarks of pseudosyllables, the attribute extraction module determines segment contours of F0 and intensity of each pseudosyllable. Subsequently, various measurements and normalization methods are used to derive the prosodic attributes (Section 3.3). The extracted prosodic attributes, which are continuous-valued, are then quantized. Prosodic tokens and various forms of n-grams are defined by one or multiple of these quantized attributes. Fixed-length term-document matrices are constructed. Some of these matrices are selected and concatenated to form a super term-document matrix (Section 4.3 and 4.4). Support vector machines are then used to build a vector-based language classifier (Section 2.5).

In this chapter, we would introduce the **generation** and **backend processing** of scores from the vector-based language classifier (Section 5.1). Then, the **evaluation metrics** for system performance will be introduced (Section 5.2). The language identification experiments reported in this chapter include two tasks. The first task is **pairwise language identification** with ten languages in the Oregon Graduate Institute Telephone Speech (OGI-TS) corpus [89] (Section 5.3). The second LID task is **language detection** in Language Recognition Evaluations (LRE) of National Institute of Standards and Technology (NIST) [14] (Section 5.4). Different constructions of term-document matrices in Chapter 4 will be tested for an optimal super term-document matrix for LID.



Target speech segments
Syllabification
Detected nuclei
ATTRIBUTE EXTRACTION
Frame estimate and segment contour derivation
Measurement
Normalization
Regression
(I) (II) (III) Raw
(I) (II) (III) Normalized
(IV) (V) (VI) (VII)
Segment contours
Prosodic attributes
— F0
- - - Intensity profile after LPF
Segment contour
segment
pair
triplet
Detected nucleus
utterance
moving window
VECTOR-BASED N-GRAM MODELING
Attribute quantization
Attribute combination
Vector space modeling of n-grams
Selection and concatenation of term-document matrices
Prosodic tokens
Unigram
n-gram
Skipping n-gram
Boundary n-gram
K documents
P terms
More frequent occurrence of the prosodic term
Counts of prosodic token unigrams from attribute t_{11}
Counts of prosodic token unigrams from attribute t_{i0}
Counts of prosodic token skipping trigrams from attribute t_{129}
Super term-document matrix from different prosodic token n-grams
VECTOR-BASED LANGUAGE CLASSIFIER
Support vector machine
Gaussian backend
Calibration
Trained models for target languages
Language recognizer decisions,
Detection errors statistics,
DET curves

Figure 5.1: *System diagram of the prosody-based language recognizer*

5.1 Detection scores from language classifier

5.1.1 Generation of detection scores

In the section, we start with a simple case where a **single detector** is used to detect the presence of a target language. Then we will proceed to a **multi-class language detector** which gives a likelihood score vector. This is the actual scenario in many LID tasks where multiple target languages are involved.

Single detector likelihood scores

A binary-class support vector machine (SVM) can readily solve a *language detection* problem for which a yes or no answer is required. It gives a likelihood score indicating the probability of the sample belonging to the target language. It uses the hyperplane equation $\boldsymbol{w}^T\boldsymbol{x}+b=0$, where $\boldsymbol{x}$ is the vector space representation of a speech segment, $\boldsymbol{w}$ and b indicate the orientation and position of the hyperplane respectively (Section 2.5). We also have y, a true or false label in the training speech segments.

In the training stage, an optimal hyperplane is found. This hyperplane can be regarded as the optimal boundary, which linearly separates the true and imposter classes $y = 1$ and $y = -1$ in the feature space. In this thesis, the features are the occurrence counts of different prosodic token n-grams.

In the testing stage, the count vector of a speech segment is processed by the support vector machine. The distance from the hyperplane, $\boldsymbol{w}^T\boldsymbol{x} + b$, is returned. The value of the distance ranges from negative infinity to positive infinity. A positive value indicates that the sample is on the side of the target class and a negative value is for the imposter class. The magnitude of the distance indicates how far the sample is from the hyperplane. If the sample is on the hyperplane, the distance is zero and it indicates equal probability to the true and the imposter classes. The distance can be used to define a log likelihood ratio, κ, where

$$\kappa^{n_t}_{\neg n_t} \equiv \log \frac{p(\boldsymbol{x}|n_t,\Gamma)}{p(\boldsymbol{x}|\neg n_t,\Gamma)}. \tag{5.1}$$

Γ is a language detector. In the case of soft margin support vector machine, $\Gamma = (\boldsymbol{w}, b, \boldsymbol{\xi})$ (Section 2.5). n_t denote the target class $y = 1$ in the support vector machine. If the prior probability is available, this likelihood ratio can be transformed by a sigmoid function to a posterior probability, based on which the binary-class classifier can make a decision.

The log likelihood ratio considering prior is given by,

$$\kappa_{p_{\neg n_t}}^{n_t} = \log \frac{p(\boldsymbol{x}|n_t, \Gamma)p(n_t)}{p(\boldsymbol{x}|\neg n_t, \Gamma)p(\neg n_t)} = \kappa_{\neg n_t}^{n_t} + \log \frac{p(n_t)}{p(\neg n_t)}. \tag{5.2}$$

Applying sigmoid function, we obtain,

$$\begin{aligned} g(\kappa_{p_{\neg n_t}}^{n_t}) &= \frac{1}{1 + \exp\left(-\kappa_{p_{\neg n_t}}^{n_t}\right)} \\ &= \left(1 + \frac{p(\boldsymbol{x}|\neg n_t, \Gamma)p(\neg n_t)}{p(\boldsymbol{x}|n_t, \Gamma)p(n_t)}\right)^{-1} = \frac{p(\boldsymbol{x}, n_t|\Gamma)}{p(\boldsymbol{x}, n_t|\Gamma) + p(\boldsymbol{x}, \neg n_t|\Gamma)} \\ &= \frac{p(\boldsymbol{x})p(n_t|\boldsymbol{x}, \Gamma)}{p(\boldsymbol{x})(p(n_t|\boldsymbol{x}, \Gamma) + p(\neg n_t|\boldsymbol{x}, \Gamma))} \\ &= p(n_t|\boldsymbol{x}, \Gamma) \qquad \text{(Posterior probability).} \end{aligned} \tag{5.3}$$

The decision of detection is made as follows,

$$p(n_t|\boldsymbol{x}, \Gamma) \geq \frac{1}{2} \quad \mapsto \text{accept to class } n_t; \tag{5.4}$$

$$p(n_t|\boldsymbol{x}, \Gamma) < \frac{1}{2} \quad \mapsto \text{reject from class } n_t. \tag{5.5}$$

In this thesis, and also conventional LID experiments, no prior information on target classes is assumed. Therefore,

$$\begin{aligned} & p(n_t) = p(\neg n_t) = \frac{1}{2} \\ \Leftrightarrow \quad & \kappa_{p_{\neg n_t}}^{n_t} = \kappa_{\neg n_t}^{n_t} \\ \Leftrightarrow \quad & g(\kappa_{p_{\neg n_t}}^{n_t}) = g(\kappa_{\neg n_t}^{n_t}). \end{aligned} \tag{5.6}$$

Apart from language detection, in the following we also include pairwise language identification experiments. This can be easily derived from the language detection case discussed above, by replacing the imposter class, $\neg n_t$, with the second target class, n_{t2}.

Multi-class likelihood score vector

Let us consider the detection problem with multiple target languages. We assume there are N target languages. For a particular speech segment, N different single-language detectors can be applied, resulting in a score vector as follows,

$$[\lambda_1 \ \ \lambda_2 \ \cdots \ \lambda_{n_t} \ \cdots \ \lambda_N]^T. \tag{5.7}$$

This vector is known as a *multi-class likelihood score vector*. In practice, λ_{n_t} can be any similarity metric. A large value of λ_{n_t} indicates high likelihood of target language n_t. It could be likelihood score, posterior probability, or likelihood ratio. In this thesis, we use *log likelihood ratio*, and Eq.(5.7) is rewritten as,

$$[\lambda_{\neg 1}^{1} \ \ \lambda_{\neg 2}^{2} \ \cdots \ \lambda_{\neg n_t}^{n_t} \ \cdots \ \lambda_{\neg N}^{N}]^T. \tag{5.8}$$

The multi-class log likelihood ratio are derived from the scaled distance κ from many single language detectors.

$$\lambda_{\neg n_t}^{n_t} = \log \frac{\exp\left(\kappa_{\neg n_t}^{n_t}\right)}{\sum_{n \neq n_t} \exp\left(\kappa_{\neg n}^{n}\right)}. \tag{5.9}$$

With no specific information, the prior probability of each class is assumed to be $p(n) = 1/N$ for all n. With reference to Eq.(5.2) and (5.3), it can be seen that the prior terms vanish and posterior probability can be calculated using the sigmoid function again. The multi-class log likelihood ratio considering prior is,

$$\begin{aligned} \lambda_{p_{\neg n_t}}^{n_t} &= \log \frac{\exp\left(\kappa_{\neg n_t}^{n_t}\right) p(n_t)}{\sum_{n \neq n_t} \exp\left(\kappa_{\neg n}^{n}\right) p(n)} \\ &= \log \frac{\exp\left(\kappa_{\neg n_t}^{n_t}\right) \frac{1}{N}}{\sum_{n \neq n_t} \exp\left(\kappa_{\neg n}^{n}\right) \frac{1}{N}} = \lambda_{\neg n_t}^{n_t}. \end{aligned} \tag{5.10}$$

Applying sigmoid function, we obtain,

$$\begin{aligned} g(\lambda_{p_{\neg n_t}}^{n_t}) &= g(\lambda_{\neg n_t}^{n_t}) = \left(1 + \exp\left(-\lambda_{\neg n_t}^{n_t}\right)\right)^{-1} \\ &= \left(\frac{\exp\left(\kappa_{\neg n_t}^{n_t}\right) + \sum_{n \neq n_t} \exp\left(\kappa_{\neg n}^{n}\right)}{\exp\left(\kappa_{\neg n_t}^{n_t}\right)} \right)^{-1} \\ &= \frac{\exp\left(\kappa_{\neg n_t}^{n_t}\right)}{\sum_{n} \exp\left(\kappa_{\neg n}^{n}\right)} \qquad \text{(Posterior probability).} \end{aligned} \tag{5.11}$$

Language detectors make decisions based on log-likelihood ratios instead of posterior probabilities. Because posterior probability is a monotonic increasing function of the log-likelihood ratio, both measures essentially give the same decision. The decision on accepting or rejecting a speech segment as in the target language n_t is made according to the following rule,

$$\lambda^{n_t}_{\neg n_t} - \theta_{n_t} \geq 0 \mapsto \text{accept to class } n_t; \tag{5.12}$$

$$\lambda^{n_t}_{\neg n_t} - \theta_{n_t} < 0 \mapsto \text{reject from class } n_t. \tag{5.13}$$

The threshold θ_{n_t} depends on the target language n_t.

5.1.2 Backend score processing

In language recognition it is allowed to perform score normalization by considering the detection scores across trials and across target languages. Suppose there are K detection trials. Consider all target languages and all trials together in an N-by-K score matrix, two backend operations are applied to improve the performance of language recognition.

The first process is known as the *Gaussian backend.* A linear discriminant analysis (LDA) transformation matrix can be applied. Geometrically, it rotates the score matrix such that the ratio of between-class variance to the within-class variance is maximized [101]. The transformation matrix is found by optimization using the score matrix of training data.

The second operation is referred to as *calibration.* It is described by the following expression,

$$\lambda'^{n_t}_{\neg n_t} = \gamma \lambda^{n_t}_{\neg n_t} + \delta_{n_t}, \tag{5.14}$$

where γ is a global scaling factor and $\boldsymbol{\delta}$ is a linear shifting vector. Optimal transformation parameters are determined from a held-out data set, subject to the maximum-a-posteriori criterion [102]. $\lambda'^{n_t}_{\neg n_t}$ is the adjusted score after calibration. Loosely speaking, calibration covers any numerical manipulation to the raw score subject to a defined criterion. More sophisticated calibration methods are discussed in Chapter 7.

5.2 Performance evaluation

All the reported LID tasks in this thesis are *closed-set detections*. Possible languages in testing trials are within a closed set known a priori. As opposed to this, *open-set detection* is a more challenging task where test data may contain unknown *out-of-set languages.*

The performance of a closed-set language recognition experiment is evaluated with a large number of input speech segments. In many studies, an evaluation metric known as *average cost performance*, C_{Avg}, is adopted. The C_{Avg} metric has been used in the NIST language recognition tasks [13, 14]. It calculates $C_{\text{detect}}(n_t)$, which is the total costs of misses and false alarms in the detector for a target language n_t. With N target languages, C_{Avg} is given by,

$$C_{\text{Avg}} = \frac{1}{N}\sum_{n_t=1}^{N} C_{\text{detect}}(n_t) \tag{5.15}$$

$$\text{where } C_{\text{detect}}(n_t) = C_{\text{Miss}} P_{\text{Target}} P_{\text{Miss}}(n_t) + \sum_{n_n \neq n_t} C_{\text{FA}} P_{\text{Non-Target}} P_{\text{FA}}(n_t, n_n). \tag{5.16}$$

C_{Miss} and C_{FA} are the penalties for the two types of errors. Typically, equal penalty of 1 is assumed. P_{Target} and $P_{\text{Non-Target}}$ is the prior probability of having the target in the detection trial. Without prior knowledge, the probability for P_{Target} is set to be $\frac{1}{2}$. The probability for having any non-target classes is $\frac{1}{2}$. In the closed-set detection with N languages, there are $N-1$ non-targets, thus probability of a particular non-target is,

$$P_{\text{Non-Target}} = \frac{1}{2}\frac{1}{N-1} \quad \text{(for closed-set detection)}. \tag{5.17}$$

With the above information, the error metric for each target language, $C_{\text{detect}}(n_t)$, is given by,

$$C_{\text{detect}}(n_t) = \frac{1}{2}P_{\text{Miss}}(n_t) + \sum_{n_n \neq n_t} \frac{1}{2}\frac{P_{\text{FA}}(n_t, n_n)}{N-1}. \tag{5.18}$$

The probability terms of errors, $P_{\text{FA}}(n_t, n_n)$ and $P_{\text{Miss}}(n_t)$, are derived from a large number of detection trials.

$$P_{\mathrm{FA}}(n_t, n_n) = P(\lambda_{\neg n_t}^{n_t} - \theta_{n_t} \geq 0 | c = n_n)$$
$$= \frac{P(\lambda_{\neg n_t}^{n_t} - \theta_{n_t} \geq 0, c = n_n)}{P(c = n_n)} = \frac{\| \mathcal{F}(n_t, n_n) \|}{\| \mathcal{I}(n_n) \|}; \tag{5.19}$$

$$P_{\mathrm{Miss}}(n_t) = P(\lambda_{\neg n_t}^{n_t} - \theta_{n_t} < 0 | c = n_t)$$
$$= \frac{P(\lambda_{\neg n_t}^{n_t} - \theta_{n_t} < 0, c = n_t)}{P(c = n_t)} = \frac{\| \mathcal{M}(n_t) \|}{\| \mathcal{I}(n_t) \|}. \tag{5.20}$$

c is the true class label. We introduce k to denote the index of a speech segment. $c(k)$ is the language of the speech segment k. In the above equations, $\mathcal{I}(n_t)$ contains the indices of speech segments whose true class is n_t. $(\mathcal{I}(n_t) : k \in [1, 2, \ldots, K] | c(k) = n_t)$. $\mathcal{F}(n_t, n_n)$ is the subset of $\mathcal{I}(n_n)$ where the indexed speech segments are falsely accepted as class n_t. $\mathcal{M}(n_t)$ is the subset of $\mathcal{I}(n_t)$ where the indexed speech segments are falsely rejected from class n_t. $\| \cdot \|$ denotes set cardinality. Physically $\| \mathcal{F}(n_t, n_n) \|$ and $\| \mathcal{M}(n_t) \|$ count respectively the number of false alarms and misses in the experimental data set.

The dominance of *detection misses* or *false alarms* in a detection experiment is affected by the detection threshold θ_{n_t}. By trying different values of θ_{n_t}, we can record the interaction between the single *miss* term and the summation of *false alarm* terms in $C_{\mathrm{detect}}(n_t)$ (Eq.(5.18)). A performance curve called *detection error tradeoff* (DET) *curve* can then be plotted.

Along the DET curve, the operating point of *equal error rate* is of great study interest. It gives the error where the term $P_{\mathrm{Miss}}(n_t)$ has the smallest difference with the weighted sum of $P_{\mathrm{FA}}(n_t, n_n)$ in $C_{\mathrm{detect}}(n_t)$.

$$C_{\mathrm{eer}}(n_t) = \underset{\theta_{n_t}}{\mathrm{eer}}\, C_{\mathrm{detect}}(n_t). \tag{5.21}$$

5.3 Pairwise language identification

Pairwise language identification with ten languages is carried out with the Oregon Graduate Institute Telephone Speech (OGI-TS) corpus [89]. Pairwise language identification accuracies will be compared with the experiments reported in literature [45, 62]. There are ten target languages in the pairwise language identification. They ımclude English, Farsi, French, German, Japanese, Korean, Mandarin, Spanish, Tamil and Vietnamese.

Table 5.1: *Paırwıse LID accuracy wıth OGI-TS 45-second speech*

	English	Farsı	French	German	Japanese	Korean	Mandarın	Spanısh	Tamıl	Vietnamese
Englısh		84 29%*	77 14%	65 71%*	82 86%	85 07%*	85 71%	90 00%	90 00%	91 43%
Farsı			80 00%*	85 71%*	90 00%*	80 60%	88 57%	67 14%*	77 14%*	90 00%
Frcnch				80 00%	94 29%	83 58%	90 00%	74 29%	94 29%	94 29%
German					85 71%	86 57%*	90 00%*	81 43%	88 57%	98 57%
Japanese						83 58%*	82 86%*	92 86%	90 00%	94 29%*
Korean							89 55%*	73 13%*	82 09%	98 51%
Mandarın								90 00%	95 71%	84 29%*
Spanısh									78 57%	91 43%
Tamıl										98 57%
Vıetnamese										
Averagc♯	83 58%	82 61%*	85 32%	84 70%	88 49%	84 74%	88 52%	82 09%	88 33%	93 49%
(cf Rouas *et al*) [62]	69 12%	70 87%	58 87%	64 60%	62 92%	68 22%	67 39%	67 18%	67 71%	62 94%
(cf Lın and Wang) [45]	81 84%	85 05%	71 51%	84 65%	86 06%	82 71%	83 41%	73 31%	76 75%	88 21%

♯ "Avcrage" ıs the accuıacy taken average from nıne paıwise LID wıth a partıculaı language They arc compaıed wıth the results fıom Rouas *et al* [62] and Lın and Wang [45]
Reported ıesults havıng lowcr accuıacıes than [62] are undeılıned
Those havıng loweı accuıacıes than [45] are mark wıth an *

Each training and test utterance is about 45 seconds long. The super term-document matrix is constructed based on the 14-attribute prosodic feature set (Section 4.5.3). Backend operations are not applıed in this experiment.

Results of the $C_2^{10} = 45$ pairwise LID experiments are included in Table 5.1. The pairwise results are compared to those reported in Rouas *et al.* [62] and Lin and Wang [45]. It is noticed that the use of the 14-attribute prosodıc feature set already gives better performance than Rouas *et al.* [62] in all language pairs but the Korean/Spanish pair. Compared with Lin and Wang [45], general performances are improved, but the language pairs involving Farsi give a worse performance.

5.4 Language detection

The second LID task is *language detectıon* in Language Recognition Evaluations (LRE) of National Institute of Standards and Technology (NIST) [14]. LRE

Table 5.2: *Target languages and channel conditions to be covered in LID model training*

Target	Channel		Target	Channel		Target	Channel	
language	VOA	CTS	language	VOA	CTS	language	VOA	CTS
Amharic	✓		Farsi	✓	✓	Portuguese	✓	
Bosnian	✓		French	✓	✓	Russian	✓	✓
Cantonese	✓	✓	Georgian	✓		Spanish	✓	✓
Creole-Haitian	✓		Hausa	✓		Turkish	✓	
Croatian	✓		Hindi	✓	✓	Ukrainian	✓	
Dari	✓		Korean	✓	✓	Urdu	✓	✓
American English	✓	✓	Mandarin	✓	✓	Vietnamese	✓	✓
Indian English		✓	Pashto	✓				

are large-scale language identification tasks. The recent LRE events were held biannually in 2003, 2005, 2007 and 2009. In the following experiments, we will include LID experiments with NIST LRE 2009. We will make comparison among different prosodic attributes. We will also compare the *full*, *skipping*, and *boundary* n-grams in terms of LID performance.

5.4.1 About NIST LRE 2009

In NIST LRE 2009, there are 23 target languages: Amharic, Bosnian, Cantonese, Creole-Haitian, Croatian, Dari, American English, Indian English, Farsi, French, Georgian, Hausa, Hindi, Korean, Mandarin, Pashto, Portuguese, Russian, Spanish, Turkish, Ukrainian, Urdu and Vietnamese [14]. Also, there is a number of candidates for the *out-of-set languages*. They are not considered as we focus on closed-set detection in this thesis.

The evaluation data includes 41793 utterances in two channel conditions, namely conversational telephone speech (CTS) and telephone bandwidth broadcast radio speech from Voice of America programmes (VOA). These utterances have nominal duration of 3, 10 or 30 seconds. The exact length of every utterance varies and no labels of nominal durations are given. Duration estimation is carried out to select 15274 30-second utterances. Then, the utterances in the out-of-set languages are removed, giving 10635 30-second utterances for the experiments reported below.

CTS Training data include NIST LRE 1996, 2003 development and evaluation sets, as well as NIST LRE 2007 development data. For broadcast radio speech, two corpora are available for training. They are from the past Voice of America radio programmes, "VOA2" and "VOA3". The total amount of data is roughly 2TB. The data is recorded in program archives from 30 minutes to 2 hours. Some training data only have possible language labels created by an automatic procedure. Such data is not used for system training in our experiments.

The broadcast radio training data contains music, repeated promotion clips, and foreign language teaching programmes. A preprocessing step of speech/music separation tries to remove them. Segmentation is also performed to extract training data with matched duration. At least 9 hours of training data is secured for each target language.

For each available language-channel condition, a target versus imposter model is trained with the binary-class support vector machine (Chapter 2.5). 34 language-channel models are trained in total and they are enumerated in Table 5.2. Data with the same language but different channels compared with the target class is removed from training. For instance, in training the Cantonese-CTS target class model, Cantonese-VOA data is discarded.

In the testing stage, CTS scores and VOA scores of the same target language will be combined (if there exist two channel models for the target language). Score combination is simply realized by the selection of larger score value among the two. Backend operations mentioned in Section 5.1.2 are applied. In *calibration*, a development set consisting of 1518 telephone speech utterances from NIST LRE 2007 evaluation set and 4523 broadcasting speech utterances from VOA3 is used [13].

$C_{\text{eer}}(n_t)$ of the 23 detectors will be found out by Eq.(5.21). An *average EER* will be computed by averaging the 23 $C_{\text{eer}}(n_t)$ figures, and comparison among different experimental conditions will be based on this figure.

Table 5.3: *Selected prosodic attributes in feature sets of different size*

Group Name	Attribute with specified normalization / extraction method		i	Attribute selected		Group Name	Attribute with specified normalization / extraction method		i	Attribute selected	
				unigram	n-gram					unigram	n-gram
(I)						(II)					
F0	Nucleus	Raw	31			Intensity	Nucleus	Raw	71		
Basic		Z-File	32	△	△	Basic		Z-File	72	△*	△*
(frame-based)		Z-File(linear)	33			(frame-based)		Z-File(linear)	73		
		Z-Utterance	34	△	△			Z-Utterance	74	△*	△*
		Z-Triplet	35	△	△			Z-Triplet	75	△*	△*
		B-File	36	△	△			B-File	76	△*	△*
		B-File(linear)	37					B-File(linear)	77		
		B-Utterance	38	△	△			B-Utterance	78	△*	△*
		B-Triplet	39	✓△	✓△♯			B-Triplet	79	✓△*	✓△*
	Maximum	Raw	41				Maximum	Raw	81		
		Z-File	42	△	△			Z-File	82	△*	△*
		Z-File(linear)	43					Z-File(linear)	83		
		Z-Utterance	44					Z-Utterance	84		
		Z-Triplet	45	△	△			Z-Triplet	85	△*	△*
		B-File	46	△	△			B-File	86	△*	△*
		B-File(linear)	47					B-File(linear)	87		
		B-Utterance	48					B-Utterance	88		
		B-Triplet	49	✓△	△			B-Triplet	89	✓△*	△*
	Minimum	Raw	51				Minimum	Raw	91		
		Z-File	52	△	△			Z-File	92	△*	△*
		Z-File(linear)	53					Z-File(linear)	93		
		Z-Utterance	54					Z-Utterance	94		
		Z-Triplet	55	△	△			Z-Triplet	95	△*	△*
		B-File	56	△	△			B-File	96	△*	△*
		B-File(linear)	57					B-File(linear)	97		
		B-Utterance	58					B-Utterance	98		
		B-Triplet	59	✓△	△			B-Triplet	99	✓△*	△*
(syllable-based)	Span	Raw	61			(syllable-based)	Span	Raw	101		
		Z-File	62	△	△			Z-File	102	△*	△*
		Z-Utterance	63	△	△			Z-Utterance	103	△*	△*
		B-File	64	✓△	△			B-File	104	△*	△*
		B-Utterance	65	△	△			B-Utterance	105	△*	△*
	Gradient	Raw	66	✓	✓		Gradient	Raw	106	✓*	✓*
		Z-File	67	△	△			Z-File	107	△*	△*
		Z-Utterance	68	△	△			Z-Utterance	108	△*	△*
		B-File	69	△	△♯			B-File	109	△*	△*
		B-Utterance	70	△	△			B-Utterance	110	△*	△*
(III)											
Duration	Nuclei	Raw	111								
Basic	Separation	Z-File	112	△	△						
(syllable-based)		Z-Utterance	113	△	△						
		B-File	114	✓△	✓△♯						
		B-Utterance	115	△	△						
	Syllable length	Raw	116								
		Z-File	117	△	△						
		Z-Utterance	118	△	△						
		B-File	119	✓△	△						
		B-Utterance	120	△	△						
	Voicing ratio	—	129	✓△	✓△♯						
(IV)						(V)					
F0	1$^{\mathrm{st}}$-order on 1 syllable		11	✓△	✓△♯	Intensity	1$^{\mathrm{st}}$-order on 1 syllable		21	△	△
regression	2$^{\mathrm{nd}}$-order on 1 syllable		12	✓△	✓△♯	regression	2$^{\mathrm{nd}}$-order on 1 syllable		22	✓△	✓△♯
	1$^{\mathrm{st}}$-order on 2 syllables		16	✓△	✓△♯		1$^{\mathrm{st}}$-order on 2 syllables		26	✓△	✓△
	2$^{\mathrm{nd}}$-order on 2 syllables		17	△	△		2$^{\mathrm{nd}}$-order on 2 syllables		27	✓△	✓△
	3$^{\mathrm{rd}}$-order on 2 syllables		18	△	△		3$^{\mathrm{rd}}$-order on 2 syllables		28	△	△♯
	4$^{\mathrm{th}}$-order on 2 syllables		19				4$^{\mathrm{th}}$-order on 2 svllables		29		
	1$^{\mathrm{st}}$-order on 3 syllables		121				1$^{\mathrm{st}}$-order on 3 syllables		125		
	1$^{\mathrm{st}}$-order on utterance		122				1$^{\mathrm{st}}$-order on utterance		126		
(VI)						(VII)					
F0	on triplet		123	✓△	✓△♯	Intensity	on triplet		127	✓△	✓△
residue	on utterance		124	△	△	residue	on utterance		128	△	△

✓ Attribute selected in 14 attribute feature set △ Attribute selected in 67-attribute feature set

♯ Attribute selected for testing different boundary n-gram

* Group (II) attributes are removed from the 14 attribute and 67-attribute, reducing to feature sets of 12 and 45 attributes

5.4.2 Comparison among different attribute groups

Recall there are seven prosodic attribute groups: (I) F0 basic, (II) Intensity basic, (III) Duration basic, (IV) F0 regression, (V) Intensity regression, (VI) F0 residue and (VII) Intensity residue (Section 3.3, Table 3.2). In Section 4.5, we use the $\overline{z_{(n_t,i)}}$ and $Isyn_{(n_t,i,j)}$ metrics for attribute selection within each group. Attributes with similar normalization/extraction methods are selected to give the 14-attribute prosodic feature set. These selected attributes are shown again in Table 5.3.

Attribute selection mentioned above is not done globally across attribute groups. It is unknown whether attributes from different groups carry complementary or redundant information. In the first experiment with NIST LRE 2009, we compare the 14-attribute feature set with seven partial sets. Each partial set has one of the attribute groups removed. The objective of this experiment is to justify the necessity to use different prosodic attributes in LID.

We also test the effect of expanding the feature set from 14 attributes to 67 attributes (Table 5.3). Various normalization methods except the raw attributes are used. The expanded feature set has 7839 terms, as opposed to the 2340 terms in the 14-attribute feature set. The objective of expanding the attribute set is to verify the working principle of "selecting attributes among similar normalization method".

Table 5.4 shows the $C_{\text{eer}}(n_t)$'s and average EER for the 14-attribute feature set and the seven partial sets. A large error in a partial set indicates the attribute group being excluded in this partial set is important. In almost all partial sets, there is an increase in average EER compared with the 14-attribute feature set. This demonstrates all prosodic attribute groups, rather than attributes of specific kind, are important to LID. Nevertheless, the EER goes down when **Intensity basic** attributes (Group (II)) are excluded. The cause of the ineffectiveness of this attribute group is unknown. It may be the poor extraction algorithm, inappropriate normalization methods or the presence of noise in this attribute group which causes the error increase. Further investigations in feature extraction and modeling will help.

Table 5.4: *EER with different attribute groups excluded (NIST LRE 2009)*

Target language	$C_{\text{eer}}(n_t)$								
	Attribute groups to exclude							14-attribute	67-attribute
	(I)	(II)	(III)	(IV)	(V)	(VI)	(VII)	feature set	featuie set
Amharic	18 59%	16 07%	18 61%	18 34%	19 34%	20 33%	18 15%	18 31%	17 08%
Bosnian	25 07%	24 56%	28 96%	26 20%	24 28%	25 08%	27 04%	26 76%	20 84%
Cantonese	12 84%	10 48%	14 20%	14 96%	13 08%	13 42%	12 80%	13 11%	11 07%
Creole-Haitian	21 00%	18 89%	21 36%	21 67%	24 16%	21 30%	18 88%	19 81%	18 88%
Croatian	25 57%	23 14%	27 37%	25 22%	25 80%	24 75%	25 53%	26 60%	24 74%
Dari	32 65%	26 25%	32 70%	34 96%	32 94%	33 93%	33 42%	32 65%	30 85%
American English	26 88%	26 41%	30 15%	29 28%	27 99%	27 19%	26 95%	27 73%	25 88%
Indian English	16 09%	15 56%	18 41%	16 42%	17 43%	16 58%	17 43%	15 06%	16 40%
Farsi	28 90%	28 39%	28 96%	30 63%	29 93%	29 17%	28 69%	27 88%	24 04%
French	20 98%	20 00%	18 74%	18 48%	18 23%	17 72%	16 96%	17 73%	17 47%
Georgian	22 05%	24 02%	25 06%	26 57%	25 56%	25 35%	25 80%	23 31%	23 57%
Hausa	15 94%	13 16%	16 95%	16 43%	14 14%	15 43%	13 66%	14 14%	12 59%
Hindi	26 24%	23 54%	23 84%	25 94%	25 19%	24 59%	22 49%	24 26%	23 24%
Korean	32 28%	25 32%	34 41%	33 33%	31 82%	33 79%	32 51%	31 82%	28 59%
Mandarin	11 38%	10 57%	12 04%	11 65%	11 55%	12 73%	11 63%	10 11%	11 28%
Pashto	21 32%	18 27%	23 14%	21 57%	21 83%	23 06%	22 80%	21 13%	19 29%
Portuguese	20 95%	20 42%	18 92%	25 93%	26 70%	21 16%	22 36%	20 93%	20 90%
Russian	26 96%	23 71%	23 71%	26 77%	24 85%	25 03%	25 62%	25 97%	22 22%
Spanish	25 42%	25 45%	26 74%	27 02%	25 97%	26 53%	23 90%	23 91%	22 80%
Turkish	19 79%	19 80%	24 93%	23 66%	21 12%	22 14%	22 14%	22 14%	20 10%
Ukrainian	38 11%	29 12%	38 91%	36 86%	36 60%	34 74%	36 64%	35 05%	36 34%
Urdu	25 58%	24 01%	25 37%	26 67%	26 62%	25 32%	24 25%	25 11%	23 51%
Vietnamese	8 81%	6 92%	7 55%	9 75%	9 43%	8 18%	7 55%	7 24%	9 12%
Average EER	22 76%	20 61%	23 52%	23 84%	23 24%	22 94%	22 49%	22 21%	20 90%

The seven attribute groups to exclude are (I) F0 basic (II) Intensity basic (III) Duration basic
(VI) F0 regression (V) Intensity regression (VI) F0 residue (VII) Intensity residue

Regarding the LID errors of the expanded feature set, a 5.9% relative reduction in average EER (from 22 21% to 20 90%) is observed by including more attributes with different normalization/extraction methods. Meanwhile, partial feature set (II) (where **Intensity basic** attributes are removed) gives even better performance in most languages. The inclusion of more attributes does not necessarily improve recognition performance. In the remaining experiments in this chapter, the **Intensity basic** attributes will be discarded.

5.4.3 The use of skipping n-grams

The use of *skipping n-grams* reduces the dimensions of term-document matrices In [45], *skipping n-grams* were also shown to give a better LID performance than full n-grams. That study used Markov models for LID, whereas the PAM-based prosodic LID system in this thesis follows the vector space modeling approach In this section, it will be shown that *skipping n-gram* models also give better LID performance in the vector-based system.

Table 5.5: *EER with full vs skipping trigrams (NIST LRE 2009)*

Model name	Number of terms♯	Average $C_{\text{cer}}(n_t)$
Full trigrams	Unigrams +2592	21.41%
Skipping trigrams	Unigrams +1296	20.61%

♯ Number of terms is the dimension of the super term-document matrix

The comparison is done in trigrams. We use the 14-attribute feature set with two **Intensity basic** attributes removed. Recall $\|\mathcal{Q}_{(i,\text{trigram})}\|$, the scalar quantization resolution of each attribute, equals 6 (Section 4.4.4). This leads to 12 (attributes) $\times 6^3 = 2592$ *full trigrams* or $12 \times (C_2^3 \times 6^2) = 1296$ *skipping trigrams.* The latter configuration gives 50% dimension reduction. LID results are shown in Table 5.5. The *skipping trigrams* with smaller dimension give better LID performance. Experiments to compare higher-order n-grams are not practical because of the exponential dimension grow of *full n-grams. Skipping n-grams* are preferred to *full n-grams* when the order of n-gram is higher than 2.

5.4.4 The use of prosodic token boundary n-grams

In this section, different *boundary n-gram* methods are compared. By using the *skipping n-grams*, prosodic token trigrams are described by some sets of $6^2 = 36$ *skipping n-grams* for the [**IGNORE**] and [**DELETE**] configurations. In [**IGNORE**], prosodic events at sentence boundaries are not distinguished from those in other positions. In [**DELETE**], prosodic events at sentence boundaries are removed. In the [**EXPLOIT**] configurations, an extra pause prosodic token is added to represent pause, giving some sets of $(6+1)^2 = 49$ distant bigrams.

We use a feature set of 10 attributes, similar to the partial set excluding **Intensity basic** attributes (Section 5.4.2). The exact attributes used for prosodic token n-gram modeling is marked in Table 5.3. We compare the language recognition EER in different boundary modeling methods. The results are shown in Table 5.6.

The [**DELETE**] configuration is the baseline. It was adopted in former experiments. The baseline configuration is justified by comparing with the

Table 5.6: *EER with different boundary n-grams (NIST LRE 2009)*

Target language	$C_{\text{eer}}(n_t)$ **IGNORE**	 **DELETE***	 **EXPLOIT**
Amharic	15.33%	14.32%	15.08%
Bosnian	27.33%	23.66%	25.07%
Cantonese	11.08%	10.32%	10.06%
Creole-Haitian	19.92%	18.32%	16.21%
Croatian	27.93%	25.31%	23.95%
Dari	29.06%	27.46%	26.19%
American English	26.23%	23.90%	24.53%
Indian English	13.28%	12.91%	13.11%
Farsi	26.92%	26.92%	23.33%
French	20.76%	18.25%	17.78%
Georgian	21.56%	24.56%	21.09%
Hausa	12.91%	14.91%	11.86%
Hindi	25.58%	23.95%	23.79%
Korean	23.59%	22.12%	20.99%
Mandarin	10.27%	10.34%	10.27%
Pashto	19.08%	18.05%	18.33%
Portuguese	20.91%	20.15%	18.60%
Russian	23.77%	22.79%	22.99%
Spanish	28.05%	25.97%	25.51%
Turkish	20.61%	20.63%	18.83%
Ukrainian	29.64%	30.15%	26.81%
Urdu	28.03%	26.33%	25.85%
Vietnamese	7.38%	6.35%	6.03%
Average EER	21.27%	20.33%	19.40%

*: [**DELETE**] resembles the baseline configuration used in previous experiments (Table 5.4, 5.5)

[**IGNORE**] configuration, which reverts to the simplest n-gram configuration with the fewest assumptions and gives a higher C_{eer}.

The [**EXPLOIT**] is the most effective to LID as it captures more information. Extra information in a *boundary n-gram* can be traced from its physical meaning. The *boundary n-gram* ($w_{s-m}\#_s$ or $\#_{s-m}w_s$) explicitly models a pseudosyllable conditioned on its position relative to the sentence boundary (#).

5.4.5 Long-range prosodic n-grams

Using the approach of *exploiting boundary n-grams*, the order of n-gram modeling is extended to 5. 5-gram modeling exhibits only marginal improvements over trigram modeling. It gives an average EER of 19.10% as compared with 19.40% with its trigram counterparts.

Chapter 6

Analysis on prosodic language identification

An LID system comprises multiple language detectors. Each detector distinguishes the target languages from many other non-target languages. In a closed-set language detection task with N languages, there are $N \times N$ terms in the confusion matrix. To further understand how the PAM-based LID system behaved, in this chapter we will break down the LID error terms and look at the **confusion between languages** (Section 6.1). Also, the PAM-based LID system makes use of a large number of prosodic attributes. The effectiveness of a certain prosodic attribute to detection is expected to vary across different target languages. We will refine the mutual information evaluation metric previously introduced in Section 4.5, and look into the **important prosodic attributes** in the detection of some target languages (Section 6.2).

6.1 LID result analysis

To analyze the LID result with prosodic features, NIST LRE 2007 is used as the testing dataset. Evaluation data consists of 2158 30-second utterances. Unlike NIST LRE 2009, channel condition of NIST LRE 2007 evaluation data is limited to conversational telephone speech (CTS). There are 21 target languages/dialects, 11 of which are dialects of a larger language class. A list of all

Table 6.1: *Target languages/dialects of NIST LRE 2007*

Target language/dialect	Target language/dialect	
Arabic	Chinese:	Cantonese
Bengali	Chinese:	Mandarin (Mainland)
Farsi	Chinese:	Mandarin (Taiwan)
German	Chinese:	Min
Japanese	Chinese:	Wu
Korean	English:	American English
Russian	English:	Indian English
Tamil	Hindustani:	Hindi
Thai	Hindustani:	Urdu
Vietnamese	Spanish:	Caribbean Spanish
	Spanish:	Non-Caribbean Spanish

languages/dialects is given in Table 6.1.

The NIST LRE 2007 dataset has only one channel condition. The proportion of data with noise, non-speech, language mislabeling is significantly lower than that in LRE 2009. These factors make LRE 2007 a preferred data set for the analysis of the relationship between different prosodic attributes and target languages. In the following, we will present the experimental results and analysis of the general language recognition task of NIST LRE 2007 [13]. Unlike the official specification, we will treat each dialect separately, and look at the performance of 21 language detectors. Also, backend score processing (Section 5.1.2) is excluded in the LID result reported in this section. This eliminates all operations across different target detectors, making the language-dependent analysis straightforward.

6.1.1 Performance evaluations on different target languages

Table 6.2 shows the $C_{\text{eer}}(n_t)$ of the 21 language detectors by using the 14-attribute feature set as introduced in Section 4.5.3. The average EER is 22.84%. From the table, it is noticed that Bengali, German, Japanese, Korean, Tamil, Vietnamese, Cantonese and Mandarin (Taiwan) can be detected with a $C_{\text{eer}}(n_t)$ below 20%. On the other hand, Russian, Urdu, Non-Caribbean Spanish and Caribbean Spanish have $C_{\text{eer}}(n_t)$ equal or above 30%.

Table 6.2: *EER of 21 detectors for NIST LRE 2007*

Target language	$C_{\text{eer}}(n_t)$ of the 14-attribute feature set
Arabic	24.71%
Bengali	17.50%
Farsi	26.26%
German	19.69%
Japanese	16.24%
Korean	18.88%
Russian	35.00%
Tamil	12.42%
Thai	23.74%
Vietnamese	14.99%
Cantonese	17.50%
Mandarin (Mainland)	21.56%
Mandarin (Taiwan)	10.39%
Min	23.90%
Wu	22.38%
American English	21.27%
Indian English	27.51%
Hindi	26.93%
Urdu	37.64%
Caribbean Spanish	29.99%
Non-Caribbean Spanish	31.10%
Average EER	22.84%

We can understand the relationship among different languages by looking at the confusion matrices. In the detection of a particular target language n_t, the error metric $C_{\text{eer}}(n_t)$ is computed by 1 term corresponding to detection miss ($P_{\text{Miss}}(n_t)$) and 20 terms corresponding to false alarms ($P_{\text{FA}}(n_t, n_n)$) (Eq.(5.18) and Eq.(5.21)). The whole confusion matrix comprises hundreds of terms. We focus on the error terms larger than 40%, which is about two times of the average EER. This gives us an idea on the worst behaving portion in the detection experiment.

The chosen error terms are listed in Table 6.3. All are false alarm terms. In the detection to target language n_t, the detector wrongly accepts a large proportion of testing samples in another language n_n. We can see a high rate of false alarms among Chinese dialects. The same observation is also found

Table 6.3: *Language pairs with high confusion rate (>40%) for NIST LRE 2007*

Target language (n_t)	Non-target language (n_n)	$P_{\mathrm{FA}}(n_t, n_n)$	Non-target language (n_n)	$P_{\mathrm{FA}}(n_t, n_n)$
[Tone languages]				
Cantonese	Mandarin (Mainland)	62.5%	Mandarin (Taiwan)	48.8%
	Wu	40.0%		
Mandarin (Mainland)	Mandarin (Taiwan)	91.0%	Min	61.3%
	Cantonese	57.5%		
Mandarin (Taiwan)	Mandarin (Mainland)	63.8%	Min	56.3%
Min*	Mandarin (Taiwan)	87.2%	Mandarin (Mainland)	75.0%
	Cantonese	57.5%		
Wu	Mandarin (Mainland)	60.0%	Cantonese	58.8%
Thai	Vietnamese	42.5%	Cantonese	41.3%
Vietnamese	Cantonese	58.8%		
[Non-tone languages]				
Arabic*	German	57.5%	Tamil	52.5%
	Non-Caribbean Spanish	50.0%		
Bengali	Urdu	40.0%		
Japanese	Korean	43.8%		
Farsi	American English	47.5%	Arabic	41.3%
Russian*	Bengali	70.0%	Urdu	55.0%
	Japanese	53.8%		
American English	Indian English	43.1%		
Indian English	Russian	56.3%	German	53.8%
	Hindi	49.4%		
Hindi	Urdu	63.8%	Indian English/Bengali	55.0%
	Tamil	45.6%		
Urdu*	Farsi	78.8%	Bengali	67.5%
	Hindi	60.6%		
Caribbean Spanish	Non-Caribbean Spanish	65.6%	Tamil	54.4%
	Bengali	53.8%		
Non-Caribbean Spanish*	Caribbean Spanish	65.0%	Tamil	54.4%
	Farsi	45.0%		

* Languages with only the three largest error terms included

in Hindustani and Spanish dialects. In the detectors of Arabic, Russian, Min, Urdu and Non-Caribbean Spanish, there are many languages giving high rate of false alarms. Only the three largest error terms are included in the table.

We are interested in the relationship between n_t-n_n for all enumerated languages. In Table 6.3, the 21 target languages are divided into two classes - tone languages and non-tone languages. Tone languages include the five Chinese dialects, Thai and Vietnamese. The remaining languages are all non-tone languages. It is noticed that all n_t-n_n pairs are purely tonal or purely non-tonal. In the following, we will look at the confusion among tone languages first, followed by that among non-tone languages. In each case, a directed graph is constructed. All arcs are directed from n_n to n_t, indicating the large likelihood of accepting the n_n samples into the n_t model.

Confusion among tone languages

Figure 6 1 is an directed graph illustrating the confusion among different tone languages In five pairs of Chinese dialects, both the forward arc and the inverted arc are present {Cantonese, Wu}, {Cantonese, Mandarin (Mainland)}, {Mandarin (Mainland), Mandarin (Taiwan)}, {Mandarin (Mainland), Min}, and {Min, Mandarin (Taiwan)} This indicates a rather symmetric confusion (i e high rate of false alarms in language n_2 in the detection of n_1, as well as in language n_1 in the detection of n_2) From the linguistic point of view, the two Mandarin dialects are closest among five dialects, and the Min dialect is spoken in the area near Taiwan These are reflected in the triangular relationship between the three dialects in the figure

By looking at Figure 6 1, we can also get a grasp of using the prosodic model for a language to detect another language For instance, Cantonese is at the tail of five arcs This indicates the possible use of Vietnamese, Thai, Min, Mandarin (Mainland) or Wu model to detect Cantonese The properties of Mandarin (Taiwan) are well modeled by the Mandarin (Mainland) model and the Min Model, resulting 91 0% and 87 2% of false alarm respectively (Table 6 3)

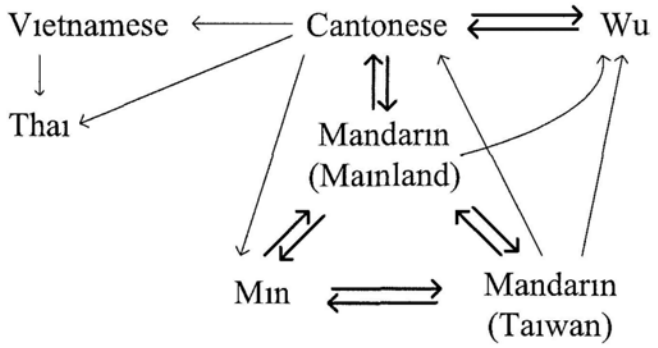


Figure 6 1 *A directed graph for tone language pairs with high confusion In the detection of a target language [head of the arc], a high rate of false alarm is found with testing samples of the confusing language [tail of the arc] Arcs are all directed from tails to heads*

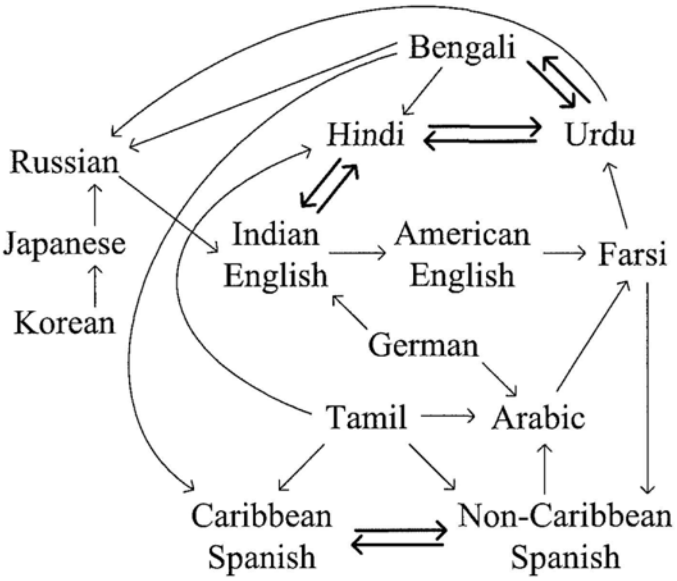


Figure 6.2: *A directed graph for non-tone language pairs with high confusion.*

Confusion among non-tone languages

There are 14 non-tone languages. The confusion pattern is more complicated than tone languages. Figure 6.2 is the directed graph for non-tone language pairs with high confusion. The Hindustani dialects, as well as the Spanish dialects, have symmetric confusion.

From Figure 6.2, connections are found among the several languages spoken in India. These languages include Bengali, Hindi, Urdu, Tamil and Indian English. Hindi is the closest with Urdu and Indian English, showing high rates of confusion. It is interesting to note a high confusion between Urdu and Bengali. Urdu is mainly spoken in Pakistan and northern India (e.g. Delhi, Jammu and Kashmir), whereas Bengali is mainly spoken in Bangladesh and eastern Indian (e.g. West Bengal).

Then we look at language pairs with a single arc. The two English dialects are good examples. The arc originates from Indian English to American English, indicating the prosodic properties of the Indian English are found in the model of the American English, and the reverse does not hold. There are two languages which have prosodic properties commonly found in the model of many other languages. In Figure 6.2, both Bengali and Tamil are at the tail of four arcs. Simply speaking, it is possible to use Arabic, Spanish and Hindi detectors to help the detection of Tamil. In Section 7.2, a related idea will be tested. Detector scores for a particular language will be used to help the detection of another "related language".

6.2 Language-specific prosodic properties

In Section 4.5, prosodic attributes are evaluated with two information-theoretic metrics, $\overline{z_{(n_t,i)}}$ and $Isyn_{(n_t,i,j)}$, in a language-independent manner. In fact, the effectiveness of a certain prosodic attribute to detection is expected to vary across different target languages. For example, F0 gradient and residue attributes are crucial to the detection of tone languages. Moving one step forward, we can look into a particular value (and its n-grams) of the attribute. A good example is the attribute of F0 gradient. Negative gradient is more indicative for detecting Mandarin than for other tone languages, due to the many occurrences of the falling-tone syllables in Mandarin. In this section, the mutual information evaluation to attributes introduced in Section 4.5.1 will be refined. Using the new metric, the relationships between different languages and prosodic attributes are revealed.

6.2.1 Bin-level mutual information evaluation

A bin-level mutual information metric is derived after $I(L_{n_t};Q_i)$ (Eq.(4.18) in Section 4.5.1). It is intended to focus on a specific value (or bin) of an attribute. We consider the entropy of L_{n_t} conditioned on the value q of attribute i, which is defined as,

$$H(L_{n_t}|Q_i\!=\!q) = -\!\!\sum_{l_{n_t}=\{0,1\}}\!\! P(l_{n_t}|Q_i\!=\!q)\log P(l_{n_t}|Q_i\!=\!q). \tag{6.1}$$

Meanwhile we extend the unigram case to consider n-gram observations of the attribute over a sequence of pseudosyllables. Let q_s denote the observed attribute value of the s^{th} pseudosyllable in an utterance. Then the entropy of target language n_t conditioned on the n-gram observation $q_{s-n+1}\cdots q_{s-1}q_s$ is given by,

$$H(L_{n_t}|q_{s-n+1}\dots q_{s-1}q_s) = -\sum_{l_{n_t}=\{0,1\}} P(l_{n_t}|q_{s-n+1}\dots q_{s-1}q_s)\log P(l_{n_t}|q_{s-n+1}\dots q_{s-1}q_s). \tag{6.2}$$

Given the limitation of data scarcity, our analysis is based on the skipping trigram [**EXPLOIT**] construction. In other words, the entropy terms cover the bigram $q_{s-1}q_s$, $q_{s-2}q_{s-1}$ and the skipping trigram $q_{s-2}q_s$ [45, 96], as well as the bigrams $q_{s-1}\#$, $\#q_s$, $q_{s-2}\#$ where $\#$ denotes an inter-utterance pause.

6.2.2 Interpretation of the metrics

Let us consider a general equation for the calculation of entropy for a binary random variable,

$$h(p) = p\log p + (1-p)\log(1-p). \tag{6.3}$$

where $p = P(L_{n_t} = 1)$ or $p = P(L_{n_t} = 1|Q_\imath = q)$. Figure 6.3 gives a plot of $h(p)$ against p (the solid line). For simplicity we consider the case that $Q_\imath$ takes only two discrete values, i.e., $\mathcal{Q}_\imath = \{1, 2\}$. First we look at the bin-level metric. $H(L_{n_t}|Q_\imath = 1)$ and $H(L_{n_t}|Q_\imath = 2)$ are obtained by substituting $p = P(L_{n_t} = 1|Q_\imath = 1)$ and $p = P(L_{n_t} = 1|Q_\imath = 2)$ respectively in Eq.(6.3). They are marked as "×" in the figure.

The geometric illustration in Figure 6.3 helps interpreting the mutual information metrics. We consider $H(L_{n_t})$, obtained by substituting $p = P(L_{n_t} = 1)$ in Eq.(6.3). The bin-level mutual information metric is a relative quantity that must be interpreted with respect to $H(L_{n_t})$. When $H(L_{n_t}|Q_\imath = q)$ lies further away from $H(L_{n_t})$, the attribute value q is considered more influential to the determination of the target language. In physical sense, we can think of an example with q corresponding to high F0. If most of the observed high-F0 pseudosyllables are from the target language, a large distance from $H(L_{n_t}|Q_\imath = q)$ to $H(L_{n_t})$ is expected. In other words, the attribute value (or bin of quantization) of "high F0" is useful in detecting the target language.

The same figure also provides some knowledge on $I(L_{n_t}; Q_\imath)$, the metric previously used in Section 4.5.1. $H(L_{n_t}|Q_i)$ is obtained as the weighted sum of the two bin-level metric, according to Eq.(4.20). Graphically, $H(L_{n_t}|Q_i)$ lies on the dashed line in Figure 6.3 and is vertically aligned with $H(L_{n_t})$. Because h is concave for $0 < p \leq 1$, $I(L_{n_t};Q_i)$ is non-negative. The attribute-level metric depends on all corresponding bins q. If the conditional entropy terms $H(L_{n_t}|Q_i = q)$ for all q's are further away from $H(L_{n_t})$, the weighted sum, $H(L_{n_t}|Q_i)$, will be smaller. $I(L_{n_t}|Q_i)$ will be larger, indicating larger amount of information contained in the attribute i about the target language.

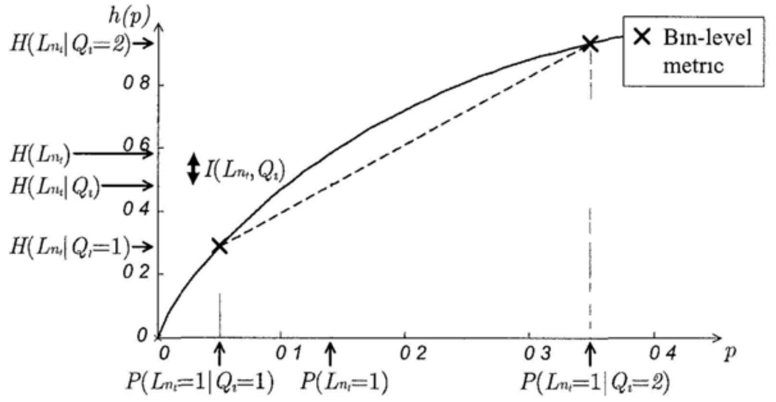


Figure 6.3: *Mutual information metrics reflected in an entropy equation*

6.2.3 Normalization of evaluation metrics

Same as $I(L_{n_t};Q_i)$, the metric previously used in Section 4.5.1, normalization is necessary for the bin-level mutual information metric. We use the identical set of random variables R to compute 500 reference conditional entropy terms $H(R|Q_i = q)$. The normalized bin-level metric is given by,

$$z_{(n_t,i,Q_i=q)} = \left|\frac{H(L_{n_t}|Q_i = q) - \mu}{\sigma}\right|, \tag{6.4}$$

where μ and σ^2 are the statistical mean and variance of $\{H(R_1|Q_i = q), H(R_2|Q_i = q), \ldots, H(R_{500}|Q_i = q)\}$. With the assumption of statistical independence between R and Q_i, the conditional entropy terms $H(R|Q_i = q)$ and the unconditional entropy terms $H(R)$ should give similar statistics of μ and σ^2. Because all R's are constructed such that $P(R = 1) = P(L_{n_t} = 1)$, μ and σ^2 can be further deduced to define an interval in which $H(L_{n_t})$ probably lies. As explained in

Section 6.2.2, the bin-level metric is a relative quantity, compared with respect to the original uncertainty of target language identity $H(L_{n_t})$.

6.2.4 Methodology

We focus on the 21 target languages in NIST LRE 2007 and calculate the normalized bin-level metric, $z_{(n_t,\imath,Q_\imath=q)}$. The speech data analyzed consists of the LRE 1996 and 2007 development sets, as well as the LRE 1996, 2003 and 2005 evaluation sets (30-second). For each language, there are at least 8000 pseudosyllables for analysis.

Under the skipping trigram [**EXPLOIT**] configuration (Section 5.4.3 and 5.4.4) with 105 attributes, for each target language there are $3 \times 105 \times 49 = 15435$ prosodic token bigrams in total for consideration. We rank the values of z metric from large to small and select the first 2500 prosodic token bigrams. Recall that the selected prosodic feature set is language-dependent. It has a dimension comparable to that of the 14-attribute feature set in Section 4.5.3. We will look at the LID results, and study the characteristics of the 2500 selected prosodic bigrams in each target language.

6.2.5 Language-specific prosodic properties

With the language-dependent prosodic feature set, the average EER is 22.26%. It is marginally better than using the language-independent 14-attribute feature set (22.84% as in Table 6.2). Another purpose of the analysis is to improve our understanding about the prosodic characteristics of different languages.

We look at the 2500 selected attribute bigrams in each target language and focus on those with large values of conditional entropy $H(L_{n_t}|q_{s-1}q_s)$. We can relate the values of conditional entropy ($h(p)$) to the probability (p) via Figure 6.3. Simply speaking, these attribute bigrams appeal to us because they have frequent occurrences in language n_t (such that the probability is large).

First let us look at Cantonese. Cantonese is a tone language with register tones [85]. Lexical tones are distinguished not only by contour shape, but also by pitch height. A unique tone sequence in Cantonese is alternating high and low tones. We consider the attribute of *F0 residue* over *Triplet* for Cantonese. Large conditional entropy is found on the bigrams $q_{s-1}q_s = \langle 15\rangle,\langle 16\rangle,\langle 51\rangle,\langle 61\rangle$ in bin-level analysis. The numbers in $\langle\cdot\rangle$ are the six-level quantized values across two pseudosyllables. For instance, $\langle 61\rangle$ indicates a high tone above the phrase curve followed by a low tone below the phrase curve. The observed value of conditional entropy reflects the abundance of L-H/H-L syllable pairs in Cantonese. This is consistent with the linguistic facts.

Then we look at other tone languages involved in this study, namely Mandarin, Vietnamese and Thai. Mandarin is similar to Cantonese, in terms of the alternating high and low tone patterns as reflected by *F0 residue* over *Triplet.* Thai and Mandarin have a popular use of falling tones. In the attribute of *F0 gradient*, the conditional entropy is large on the bigrams $\langle 1*\rangle$ and $\langle *1\rangle$ where $*$ means any attribute value. The smallest attribute value "1" denotes a falling tone. Vietnamese shows the opposite trend. Conditional entropy in *F0 gradient* is large for the bigrams $\langle 6*\rangle$ and $\langle *6\rangle$. This indicates the abundance of rising tones in the language.

Japanese is a pitch-accent language. There are only two pitch levels (H and L) in the language. In each word, one accented syllable (or mora in a rigorous sense) is found at most. There is a pitch transition from H to L immediately after the accented mora. In other morae the pitch stays flat as either H or L [103]. Due to this linguistic fact, the conditional entropy of *F0 span* (*B-File*) for Japanese stands out. The bigrams $\langle 15\rangle,\langle 51\rangle,\langle 16\rangle$ and $\langle 61\rangle$ are believed to indicate the pitch transition before or after an accented mora. They give high conditional entropy. Other distinctive bigrams include $\langle 11\rangle,\langle 12\rangle$ and $\langle 21\rangle$. They indicate the unaccented part of the word where the pitch stays flat and the vertical span of F0 is minimal.

Apart from tone and pitch-accent languages discussed above, language-specific properties can also be found in non-tone languages. For example, Tamil is a language with no lexical stress [104]. This linguistic fact is reflected by the regular rhythm and intensity observed in the *intensity nucleus*, *residue* and *nuclei separation* (*B-File*) attributes.

Chapter 7

Score fusion and calibration

Various LID experiments are reported in Chapter 5. These results are generated by an LID system solely based on prosodic features. In the first part of this chapter, we will show optimal ways to **combine the results of the PAM-based prosodic LID system and the state-of-the-art phonotactic LID system**. LID performance enhancements will be demonstrated. In the second part of this chapter, some **numerical adjustment to scores** will also be carried out, which brings further improvements to system results.

7.1 Application-independent score fusion

Fusion is the combination of multiple sets of scores. It is common in a large-scale LID system. In PPRLM systems, multiple set of LID scores are generated from parallel streams of tokenizers [105]. In other cases, multiple sets of scores are from different LID sub-systems. In [106], LID sub-systems with acoustic, phonotactic and prosodic features were stacked together. Log scores of different systems were added together to give the score of an ensemble classifier. In [74], Gaussian backend process (Section 5.1.2) was incorporated in the score fusion of GMM-LM, PPRLM and GMM-SVM systems. In [45], fusion method was derived in a Bayesian framework [107]. In general, fusion methods are implemented by linearly combining the component scores. The difference lies on the method to derive the combination weights.

In the following, the PAM-based prosodic LID system is fused with a state-of-the-art phonotactic LID system on the score level The phonotactic LID system adopts a parallel phone recognition followed by vector space model (PPRVSM) approach [108] It is one of the subsystems in the Institute for Infocomm Research submission to NIST LRE 2009 Compared with the average EER in the PAM-based prosodic LID system, which is above 20%, the PPRVSM system has an average EER below 5%

Let ${\lambda_1}^{n_t}_{\neg n_t}$ and ${\lambda_2}^{n_t}_{\neg n_t}$ be scores from the two systems *Fusion* is carried out by extending the *calibration* equations (Eq (5 14))[102] The fused score is,

$$\lambda'^{n_t}_{\neg n_t} = \gamma_1 {\lambda_1}^{n_t}_{\neg n_t} + \gamma_2 {\lambda_2}^{n_t}_{\neg n_t} \tag{7 1}$$

Optimal weights γ_1 and γ_2 are found by an objective function subject to the maximum posterior probability (Eq (5 11)) This objective does not assume any particular LID scenario It is independent with parameters such as C_{Miss}, C_{FA} and prior probabilities in the calculation of detection costs Therefore, it is known as *application-independent fusion* [102]

The linear translation term δ_{n_t} is dropped in score fusion Intuitively, the target-dependent bias should have been handled by *backend calibration* so this extra term is no longer necessary We compare the two cases with and without δ_{n_t} in our fusion experiment For two sub-systems which behave differently in terms of $C_{\text{eer}}(n_t)$, the omission of δ_{n_t} gives better results

LID experiments reported in Chapter 5 for NIST LRE 2009 will be repeated with score fusion Development and testing sets for both systems are identical The training set for the PPRVSM system include more speech data from the CALLFRIEND and OHSU Corpora

7.1.1 Score fusion with a phonotactic LID system

Table 7 1 compares the results of different fusion scores in NIST LRE 2009 On the leftmost column is the result of the state-of-the-art phonotactic LID system Its average EER is 3 56% The next column shows the errors from

Table 7 1 *EER when different prosodic feature sets are fused with PPRVSM in the score level (NIST LRE 2009)*

Target language	$C_{\text{eer}}(n_t)$ for Phonotactic PPRVSM system	$C_{\text{eer}}(n_t)$ after fusion with			
		14-attribute set	7 partial sets[♯]	Partial set (II)[§]	
				[DELETE]	[EXPLOIT]
Amharic	0 75%	0 54%	0 76%	0 75%	0 51%
Bosnian	9 30%	8 10%	7 90%	7 04%	7 32%
Cantonese	1 56%	1 31%	1 31%	1 06%	1 32%
Creole Haitian	2 11%	2 12%	1 55%	1 55%	1 55%
Croatian	5 61%	5 60%	6 07%	5 80%	6 12%
Dari	8 74%	8 18%	7 98%	8 00%	8 02%
American English	3 73%	4 17%	3 75%	3 91%	4 11%
Indian English	5 24%	3 89%	4 23%	4 51%	4 52%
Farsi	1 99%	2 09%	2 05%	2 29%	2 05%
French	2 79%	2 49%	2 50%	2 26%	2 27%
Georgian	1 54%	1 51%	1 47%	1 50%	1 30%
Hausa	1 28%	1 03%	0 79%	1 00%	0 73%
Hindi	8 40%	7 50%	7 80%	7 23%	7 26%
Korean	1 30%	1 08%	0 63%	0 84%	0 70%
Mandarin	1 15%	1 16%	1 26%	1 08%	1 08%
Pashto	4 77%	3 55%	3 55%	3 29%	3 26%
Portuguese	1 26%	1 28%	1 51%	1 46%	1 30%
Russian	2 33%	2 83%	2 48%	2 73%	2 75%
Spanish	1 54%	1 30%	1 24%	1 30%	1 30%
Turkish	1 27%	0 79%	0 82%	0 79%	0 80%
Ukrainian	6 67%	5 67%	5 93%	5 96%	5 45%
Urdu	5 81%	5 78%	5 54%	5 27%	5 32%
Vietnamese	2 83%	2 20%	1 94%	3 10%	1 91%
Average EER					
(before fusion)	3 56%	22 21%	—	20 33%	19 40%
(after fusion)	—	3 22%	3 18%	3 10%	3 08%

[♯] 7 partial sets are formed by excluding attributes in different prosodic groups

[§] Partial set (II) is the 14 attribute set with *Intensity basic* attributes excluded

the phonotactic-prosodic fused score where 14 prosodic attributes introduced in Section 4 5 3 are used Score fusion brings a 9 6% relative reduction of average EER to 3 22%

In the third column, seven sets of scores from all partial prosodic sets in Section 5 4 2 are used Together with the PPRVSM phonotactic scores, a score fusion with eight systems is carried out In Table 5 4, LID performance of the seven partial sets was shown to vary Score fusion from multiple sets of prosodic scores elicits complementary effects The average EER, 3 18%, is smaller than that using 14 prosodic attributes altogether

In the two rightmost columns in Table 7 1, the partial set (II) with **Intensity**

basic attributes excluded is used with different configurations of boundary n-grams. In Section 5.4.4 and Table 5.6, these constructions demonstrate better performance over the basic setup. These trends of EER in the prosodic LID system before fusion are shown to be brought forward to the fused system with the PPRVSM scores in Table 7.1.

In the state-of-the-art LID, acoustic/phonotactic systems generally perform much better than prosodic systems. Nevertheless, score-level fusion trials with different prosodic systems do show EER reduction from 9.6% to 13.5% relatively. From the results in Table 7.1, it seems to be the case if a prosody-based system has a lower EER, the fusion result will also be better.

There have been disputes whether prosodic features can assist in a LID task in general. A common belief is that the use of prosodic features is language-specific [10, 105]. We test the four tonal languages (Cantonese, Hausa, Mandarin and Vietnamese) and four other languages (Amharic, Indian, French and Turkish) which give smallest EER with the 14-attribute prosodic set. Significant EER reductions are observed for these languages after score fusion with the PPRVSM system. If only the test data of these eight languages are retained, prosodic features bring a 30.9% relative reduction of EER, from 1.81% to 1.25%.

7.2 Target dependent score calibration

While prosodic features bring significant EER reductions to some languages, there are some "difficult" languages whose EER is high for both PPRVSM and prosodic systems. Among the 23 target languages in NIST LRE 2009, five related language pairs are generally considered to be mutually intelligible [14]. They are listed below,

- Russian-Ukrainian
- Bosnian-Croatian
- Hindi-Urdu
- English(American)-English(Indian)
- Farsi-Dari

Public results reveal higher recognition errors in these related pairs [9]. Detection to these related languages becomes a bottleneck in a state-of-the-art language recognition system. Intuitively, if some error reduction techniques specific to these related languages are introduced, there is hope to reduce the global error.

Let n_1 and n_2 represent two related languages. They are mutually intelligible. Detection among n_1 and n_2 is believed to give many *misses* and *false alarms*. Here we make two hypotheses:

Hypothesis 1: Cost minimization specific to n_1 and n_2 would be beneficial to the reduction of the global cost performance.

Hypothesis 2: The log likelihood ratios for n_1 and n_2 contain similar and complementary information.

We will start with a set of language recognition results which is calibrated in the global level. Further calibration specific to those related language pairs are conducted.

7.2.1 Evaluation metrics

It is important to review the evaluation metrics before some calibration methods are proposed to reduce errors. In the calculation of average cost performance in NIST Evaluations, detection scores to all target languages are pooled together. After pooling, a single detection threshold is used for detection. The decision making process is rewritten by replacing the language-dependent threshold θ_{n_t} introduced in Section 5.1,

$$\lambda^{n_t}_{\neg n_t}(k) - \theta \geq 0 \mapsto \text{accept } k \text{ belongs to class } n_t; \tag{7.2}$$

$$\lambda^{n_t}_{\neg n_t}(k) - \theta < 0 \mapsto \text{reject } k \text{ belongs to class } n_t. \tag{7.3}$$

The error terms in the pooled data are,

$$P_{\text{FA}}(n_t, n_n) = P(\lambda^{n_t}_{\neg n_t} - \theta \geq 0 | c = n_n) = \frac{\| \mathcal{F}(n_t, n_n) \|}{\| \mathcal{I}(n_n) \|}; \tag{7.4}$$

$$P_{\text{Miss}}(n_t) = P(\lambda^{n_t}_{\neg n_t} - \theta < 0 | c = n_t) = \frac{\| \mathcal{M}(n_t) \|}{\| \mathcal{I}(n_t) \|}. \tag{7.5}$$

Physically, $\| \mathcal{F}(n_t, n_n) \|$ and $\| \mathcal{M}(n_t) \|$ count the number of false alarms and misses in the experimental data set. An example of detection likelihood of

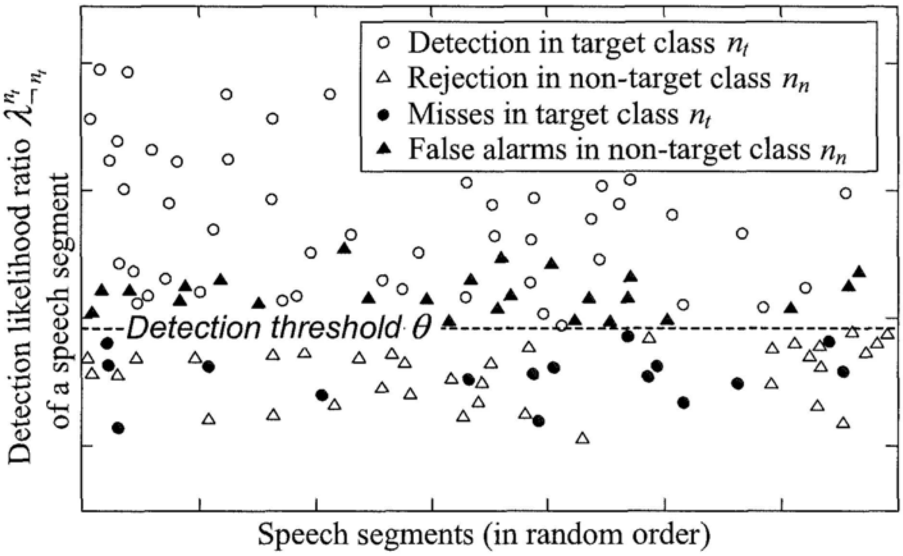


Figure 7.1: *Likelihood ratio* $\lambda^{n_t}_{\neg n_t}$ *for* n_t *detection in a data set with two classes:* n_t*,* n_n

a two-class data set with target class n_t and non-target class n_n is plotted in Figure 7.1, in which $\|\mathcal{M}(n_t)\|$ can be obtained by counting the number of filled circles, while $\|\mathcal{F}(n_t, n_n)\|$ is the number of filled triangles.

The dominance of *detection miss* or *false alarms* in the detection experiments is affected by a global detection threshold θ. By varying the value of θ, there will be different *operating points*. The *pooled EER* is the error at the operating point where the weighted sum of all $P_{\text{Miss}}(n_t)$ terms has the smallest difference with the weighted sum of $P_{\text{FA}}(n_t, n_n)$ in all language pairs. It is denoted by,

$$C_{\text{p-eer}} = \underset{\theta}{\text{eer}}\ C_{\text{Avg}}, \tag{7.6}$$

where C_{Avg} is given by,

$$C_{\text{Avg}} = \frac{1}{N}\sum_{n=1}^{N} C_{\text{detect}}(n_t), \tag{7.7}$$

$$C_{\text{detect}}(n_t) = \frac{1}{2}P_{\text{Miss}}(n_t) + \sum_{n_n \neq n_t} \frac{1}{2}\frac{P_{\text{FA}}(n_t, n_n)}{N-1}. \tag{7.8}$$

Another operating point in our interest is the minimum global average cost, which is defined by,

$$C_{\text{min}} = \min_{\theta}\ C_{\text{Avg}}. \tag{7.9}$$

Note that both C_{min} and $C_{\text{p-eer}}$ are based on a global threshold across all languages. This is different from the C_{eer} metrics used in earlier chapters.

7.2.2 Problem formulation

Cost minimization by likelihood ratio adjustment

Consider two related languages n_1, n_2. According to Hypothesis 1, we propose to minimize the cost terms C_{n_1,n_2} and C_{n_2,n_1}, where

$$C_{n_1,n_2} = P_{\text{Miss}}(n_1) + \frac{1}{N-1} P_{\text{FA}}(n_1, n_2), \tag{7.10}$$

$$C_{n_2,n_1} = P_{\text{Miss}}(n_2) + \frac{1}{N-1} P_{\text{FA}}(n_2, n_1). \tag{7.11}$$

Eq.(7.10) and (7.11) are rewritten forms of Eq.(7.8), retaining only the cost components related to classes n_1 and n_2. Note the cost for a single detection miss is $N-1$ times of the cost for a single false alarm. This ratio is inherited from the C_{detect} definition in Eq.(7.8).

In the following, the minimization of C_{n_1,n_2} is illustrated as an example. Let $n_t = n_1$ be the *target language* and $n_r = n_2$ is the *related language*. Referring to Eq.(7.4) and (7.5), we can choose to adjust the threshold θ and/or the likelihood ratio $\lambda^{n_t}_{\neg n_t}$ for a smaller C_{n_t,n_r}. Because this cost minimization is specific to n_t and n_r only, we fix the global parameter θ and adjust $\lambda^{n_t}_{\neg n_t}$.

Another issue is that target class specific cost minimization should be performed to the in-class data in n_t or n_r only, while this information is generally unavailable in the testing set. The workaround is to use a rough estimate of target class. Let $\tilde{\mathcal{I}}(n)$ be the estimated indices of speech segments in language n (i.e. estimate of $\mathcal{I}(n)$). $\tilde{\mathcal{I}}(n)$ is derived heuristically. By evaluating the vector of detection likelihood ratios of speech segment k (Eq.(5.8)), k is put in $\tilde{\mathcal{I}}(n)$ if $\lambda^{n}_{\neg n}(k)$ is found to be among the largest three ratios.

In cost minimization, the goal is to have an adjusted $\lambda'^{n_t}_{\neg n_t}$ such that both sets $\mathcal{M}(n_t)$ and $\mathcal{F}(n_t, n_r)$ shrink. According to Hypothesis 2, $\lambda^{n_t}_{\neg n_t}$ and $\lambda^{n_r}_{\neg n_r}$ contain similar and complementary information. We propose the following adjustment,

$$\lambda'^{n_t}_{\neg n_t}(k, \alpha_{n_t,n_r}) = \lambda^{n_t}_{\neg n_t}(k) + \tilde{\tau}_{n_t,n_r}(k, \alpha_{n_t,n_r}),$$

where

$$\tilde{\tau}_{n_t,n_r}(k, \alpha_{n_t,n_r}) = \begin{cases} \alpha_{n_t,n_r}\lambda^{n_r}_{\neg n_r}(k) & \text{if } k \in \{\tilde{\mathcal{I}}(n_t) \cup \tilde{\mathcal{I}}(n_r)\}. \\ 0 & \text{otherwise.} \end{cases} \tag{7.12}$$

Literally, Eq.(7.12) says that in the detection of language n_t, the log likelihood ratio for a subset of speech segment indexed $k \in \{\tilde{\mathcal{I}}(n_t) \cup \tilde{\mathcal{I}}(n_r)\}$ has to be adjusted as a linear combination of $\lambda^{n_t}_{\neg n_t}(k)$ and $\lambda^{n_r}_{\neg n_r}(k)$. $\lambda'^{n_t}_{\neg n_t}$ is the adjusted likelihood ratio. α_{n_t,n_r} is the weight for likelihood ratio combination.

After $\lambda^{n_1}_{\neg n_1}$ is adjusted for the minimization of cost C_{n_1,n_2}, minimization of C_{n_2,n_1} can be done in the same manner. By substituting $n_t = n_2$ and $n_r = n_1$ and repeating the operation in Eq.(7.12), $\lambda'^{n_2}_{\neg n_2}$ is found from $\lambda^{n_2}_{\neg n_2}$ and $\tilde{\tau}_{n_2,n_1}$.

Optimal parameters for score calibration

For the likelihood ratio adjustment of each *target language* n_t with its *related language* n_r, we use the development data set to find the optimal parameters such that the errors indicated by Eq.(7.4) and Eq.(7.5) are minimized. Instead of minimizing the sets $\|\mathcal{F}(n_t, n_r)\|$ and $\|\mathcal{M}(n_t)\|$, we propose to minimize the *total erroneous deviations* of likelihood ratios in the development data set. *Erroneous deviations* can be easily visualized in Figure 7.1. For a detection miss, it is the vertical distance from the detection threshold θ to the filled circle. For a false alarm, it is the vertical distance from the filled triangle to θ. Mathematically, the minimization of *total erroneous deviations* is formulated as,

$$\begin{aligned} &\min_{\upsilon,\alpha_{n_t,n_r}} \quad \sum_{k=1}^{K} \max\left(y_{n_t}(k) \times f(k, \alpha_{n_t,n_r}, \upsilon), 0\right) \\ &\text{subject to (s.t.)} \quad |\alpha_{n_t,n_r}| \leq 1, \\ &f(k, \alpha_{n_t,n_r}, \upsilon) = \lambda'^{n_t}_{\neg n_t}(k, \alpha_{n_t,n_r}) - (\theta + \upsilon), \\ &y_{n_t}(k) = \begin{cases} -(N-1) & \text{if } k \in \mathcal{I}(n_t). \\ 1 & \text{otherwise.} \end{cases} \end{aligned} \tag{7.13}$$

$f(\cdot)$ is the *deviation* of $\lambda'^{n_t}_{\neg n_t}$ from a reference point $(\theta+\upsilon)$. $\lambda'^{n_t}_{\neg n_t}$ is the adjusted likelihood ratio defined in Eq.(7.12). The reference point is the detection threshold θ, shifted by υ. $y_{n_t} \times f(\cdot)$ returns positive values for erroneously detected segments and negative values for appropriately detected ones. The $\max(\cdot)$ operation removes *deviations* which are not erroneous. Among the positive-valued *deviations* there are two error types: misses and false alarms. y_{n_t} scales the two error types with the default $N-1:1$ ratio. Every α_{n_t,n_r} is bounded such that $\lambda'^{n_t}_{\neg n_t}$ lies in a suitable range. The objective function is convex on α_{n_t,n_r}. Thus, with a fixed υ, a globally optimal solution of α_{n_t,n_r} can be found [109]. The objective function in Eq.(7.13) tries to push the likelihood ratios of detection misses and false alarms towards the reference point $(\theta+\upsilon)$.

The polarity of υ indicates the optimization goal towards fewer misses or fewer false alarms. Referring to Figure 7.1, a positive υ pushes the dashed line (reference point) upwards. Thus there will be more filled circles (missed targets) included in the optimization in Eq.(7.13), and the parameter for likelihood ratio adjustment, α_{n_t,n_r}, will be optimized towards the goal of having fewer misses. Oppositely, a negative υ will lead to an optimal parameter which favours fewer false alarms.

Experimental procedures

Detection target dependent calibration is carried out for each of the five related language pairs, namely Bosnian and Croatian, Dari and Farsi, Hindi and Urdu, Russian and Ukrainian, American and Indian English. These language pairs are highlighted in the NIST LRE 2009 task specification [14]. We do not propose any methods in finding out these pairs.

Figure 7.2 shows the system diagram of the complete language detection system, in which the two shaded blocks are the modules of *detection target dependent calibration* for one pair of related target languages n_1 and n_2. A pair of adjusted likelihood ratios, denoted by $\lambda'^{n_1}_{\neg n_1}$ and $\lambda'^{n_2}_{\neg n_2}$, is derived. There are totally ten target languages (in five pairs) whose likelihood ratios are ad-

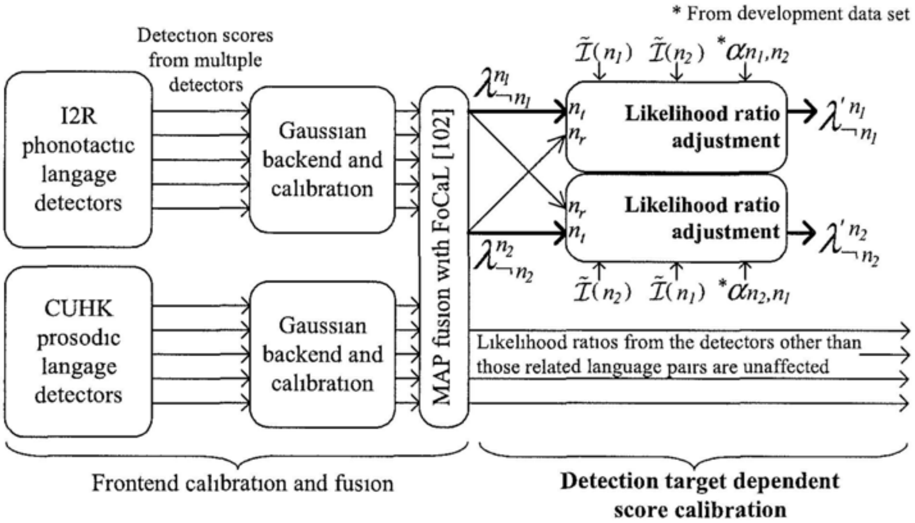

* From development data set
Detection scores from multiple detectors
I2R phonotactic langage detectors
CUHK prosodic langage detectors
Gaussian backend and calibration
Gaussian backend and calibration
MAP fusion with FoCaL [102]
$\lambda^{n_l}_{\neg n_l}$
$\lambda^{n_2}_{\neg n_2}$
$\tilde{\mathcal{I}}(n_l)$ $\tilde{\mathcal{I}}(n_2)$ $^*\alpha_{n_l,n_2}$
n_l
n_r
Likelihood ratio adjustment
$\lambda'^{n_l}_{\neg n_l}$
n_r
n_l
Likelihood ratio adjustment
$\lambda'^{n_2}_{\neg n_2}$
$\tilde{\mathcal{I}}(n_2)$ $\tilde{\mathcal{I}}(n_l)$ $^*\alpha_{n_2,n_l}$
Likelihood ratios from the detectors other than those related language pairs are unaffected
Frontend calibration and fusion
Detection target dependent score calibration

Figure 7.2: *System diagram of score calibration*

justed following Eq.(7.12). In each adjustment, the optimal α_{n_t,n_r} is found from a development data set, with the objective function in Eq.(7.13). A convex optimization tool, cvx, is used [110]. The optimal α_{n_t,n_r} parameters are then substituted in Eq.(7.12) with the NIST LRE 2009 evaluation set. C_{min} and $C_{\text{p-eer}}$ with the evaluation data is reported.

We focus on an application-independent fusion system with PPRVSM and seven partial sets (Table 7.1). For notation simplicity, scores and error costs after the first-pass global application-independent fusion are referred to as *original* scores hereinafter. On top of the original scores, detection target dependent score calibration is carried out to obtain *calibrated* scores.

7.2.3 Experimental results

Reference point for erroneous deviation minimization

To test the best reference point $(\theta + \upsilon)$ in Eq. (7.13), the experimental procedures described above are repeated with different values of υ. Recall that a positive υ favours fewer misses and a negative υ favours fewer false alarms. A sequence of υ from -6 to 6, spaced 0.5 apart, is tested. C_{min} and $C_{\text{p-eer}}$ with the evaluation data are plotted in Figure 7.3.

From the figure, a clear trend of increasing errors can be observed if the

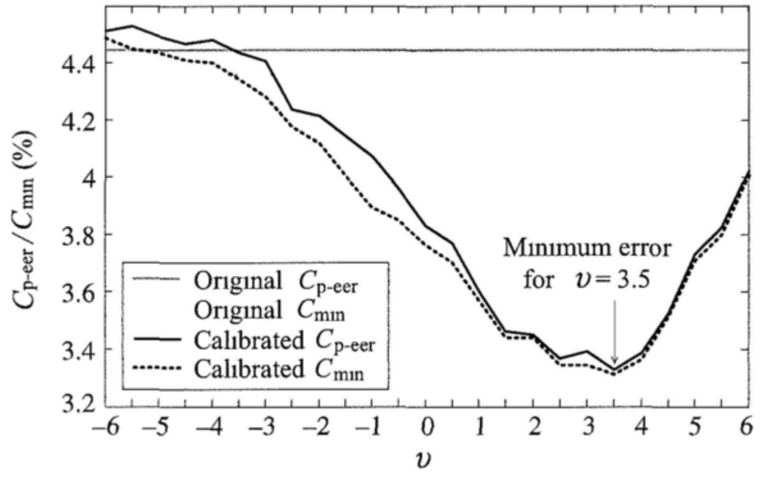


Figure 7.3: *Original and calibrated C_{min}, $C_{p\text{-}eer}$ with optimal α_{n_t,n_r} under different values of v*

value of v is too large or too small. Both $C_{\min}$ and $C_{\text{p-eer}}$ attain the lowest values when v equals 3.5. This positive value implies optimization of parameter α in Eq.(7.13) should prefer fewer detection misses. It is reasonable since the error cost for a detection miss is $N - 1$ times of the cost for a false alarm, as defined in Eq.(7.10) and (7.11).

The exact value of v depends on the *erroneous deviations* of likelihood ratios, which are the vertical distances between detection threshold θ and filled circles/triangles in Figure 7.1. By inspecting the scores with the development data set, *erroneous deviations* of likelihood ratios are generally found to be smaller than 6. Therefore v is tried in the range from −6 to 6. Unique optimal v can also be trained in the optimization for different n_t, n_r pairs. Nevertheless, a reasonable guess of a universal v already leads to $C_{\min}$ and $C_{\text{p-eer}}$ reduction, compared with the original error terms as shown in the horizontal lines in Figure 7.3.

Global detection errors

Figure 7.4 shows the detection errors for the evaluation data with original and calibrated scores. Original scores are those globally calibrated with FoCaL [102]. Calibrated scores are obtained via the proposed calibration method. v is chosen to be 3.5. $C_{\min}$ and $C_{\text{p-eer}}$ for the original scores over 23 target languages is 4.36%

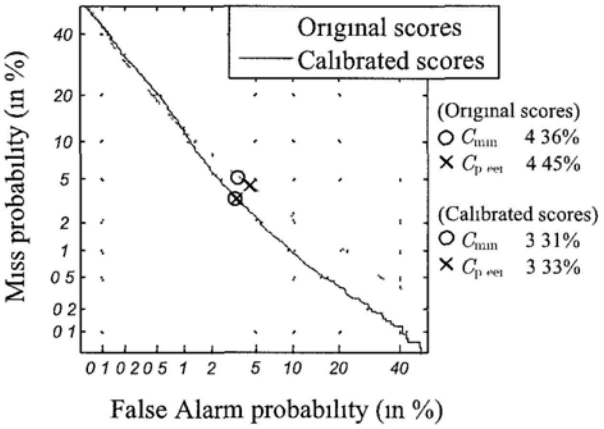


Figure 7.4: *DET plot for the original and the calibrated scores with* $v = 3.5$

and 4.45% respectively. After the proposed calibration, $C_{\min}$ and $C_{\text{p-eer}}$ over 23 target languages are reduced to 3.31% and 3.33% respectively. A relative EER reduction of 25.2% is achieved.

Table 7.2 shows the error statistics at the global $C_{\min}$ and $C_{\text{p-eer}}$ operating points for every target language. Results of the five related language pairs are enumerated separately to the top of the table. As expected, the major contribution in error reduction comes from the five related language pairs. The 10 languages except Russian have error reductions after calibration. For the other 13 languages, although their detection scores are untouched, reduction in $C_{\min}$ and $C_{\text{p-eer}}$ can also be observed.

Detection errors in different target languages

Detection target dependent calibration is shown to reduce the pooled EER in the global data set. In this section, the source of error reduction in the five pairs of related languages will be investigated.

Referring to Eq.(7.8), the component terms of $C_{\text{detect}}(n_t)$ are recorded in Table 7.3 for analysis. These components include the miss rate to a target language ($P_{\text{Miss}}(n_t)$), and the overall false alarm rate ($\sum \frac{P_{\text{FA}}(n_t,n_n)}{N-1}$). They summarize the detection of target language n_t from all imposter languages, and are referred to as *global error terms* hereinafter.

It is reminded at the *pooled EER* operating point, the *global error terms* of

Table 7.2: *Error statistics before and after score calibration*

	Before calibration		After calibration	
	(With $\lambda^{n_t}_{\neg n_t}$)		(With $\lambda'^{n_t}_{\neg n_t}$)	
	C_{min}	$C_{\text{p-eer}}$	C_{min}	$C_{\text{p-eer}}$
Bosnian	20.25%	18.54%	8.11%	8.12%
Croatian	7.78%	6.92%	6.45%	6.48%
Dari	8.97%	9.07%	7.14%	7.03%
Farsi	3.21%	3.67%	2.60%	2.65%
American English	3.86%	4.00%	3.57%	3.61%
Indian English	4.21%	4.53%	3.75%	3.79%
Hindi	7.88%	8.43%	5.42%	5.46%
Urdu	5.77%	6.61%	5.29%	5.35%
Russian	4.15%	5.21%	5.28%	5.35%
Ukrainian	10.89%	9.90%	6.39%	6.40%
Average of 5 related language pairs	7.70%	7.69%	5.40%	5.42%
Amharic	1.09%	1.34%	0.88%	0.89%
Cantonese	1.30%	1.34%	1.35%	1.36%
Creole-Haitian	1.65%	1.91%	1.79%	1.81%
French	2.44%	2.74%	2.24%	2.28%
Georgian	1.38%	1.55%	1.49%	1.49%
Hausa	0.81%	0.91%	0.81%	0.84%
Korean	0.71%	0.96%	0.56%	0.57%
Mandarin	1.44%	1.46%	1.28%	1.29%
Pashto	3.56%	4.11%	3.52%	3.46%
Portuguese	1.54%	1.63%	1.41%	1.44%
Spanish	2.90%	3.87%	2.21%	2.26%
Turkish	2.41%	1.56%	2.64%	2.65%
Vietnamese	2.04%	1.99%	2.00%	2.02%
Average of other 13 languages	1.79%	1.95%	1.71%	1.72%
Average on 23 languages	4.36%	**4.45%**	3.31%	**3.33%**

miss and false alarm probabilities in a single target language do not have to satisfy the *equal error* criterion. For instance, with original scores, P_{Miss} and P_{FA} for Bosnian is 35.49% and 1.58% at the *pooled EER* operating point, giving C_{detect} of 18.54% (Table 7.2).

As opposed to the *global error terms*, the false alarm rate specific to a related language pair ($P_{\text{FA}}(n_t, n_r)$) is also included in Table 7.3. This specific false alarm term indicates how well the LID system can classify the two related languages before and after calibration. The optimal parameter α_{n_t,n_r} found by Eq.(7.13) is also recorded in Table 7.3.

α_{n_t,n_r} specifies the proportion of the related language likelihood ratio ($\lambda^{n_r}_{\neg n_r}$)

Table 7.3: *Pooled EER for different targets*

n_t (Target language)	$C_{\text{p eer}}$ before calibration [Global error terms] $P_{\text{Miss}}(n_t)$	$\sum\frac{P_{\text{FA}}(n_t,n_n)}{N-1}$[♭]	$P_{\text{FA}}(n_t,n_r)$[♯]	α_{n_t,n_r}	$C_{\text{p eer}}$ after calibration [Global error terms] $P_{\text{Miss}}(n_t)$	$\sum\frac{P_{\text{FA}}(n_t,n_n)}{N-1}$[♭]	$P_{\text{FA}}(n_t,n_r)$[♯]
Bosnian	35.49%	1.58%	23.94%	0.76	12.68%	3.57%	71.28%
Croatian	8.78%	5.07%	74.93%	0.43	8.78%	4.18%	79.72%
Dari	14.91%	3.22%	14.32%	0.34	11.05%	3.01%	35.29%
Farsi	0.26%	7.08%	72.49%	−0.30	0.51%	4.79%	47.81%
American English	2.08%	5.92%	55.50%	0.05	3.40%	3.81%	45.52%
Indian English	2.54%	6.51%	38.93%	0.13	3.05%	4.54%	37.50%
Hindi	4.20%	12.67%	80.74%	0.62	1.80%	9.13%	97.89%
Urdu	2.11%	11.12%	85.76%	0.67	2.11%	8.58%	96.85%
Russian	0.00%	10.43%	52.06%	−0.27	0.19%	10.52%	43.56%
Ukrainian	19.07%	0.73%	3.25%	0.76	10.82%	1.98%	34.03%

[♯] n_r is the related language (i.e. the other language in the pair)

[♭] n_n includes all languages other than n_t

to be added to the target language likelihood ratio ($\lambda^{n_t}_{\neg n_t}$) in calibration (Eq.(7.12)). By looking at the parameter α_{n_t,n_r}, two scenarios can be observed.

In the first scenario, a negative α_{n_t,n_r} is found to be optimal. Take Russian detection as an example and refer to Eq.(7.12), such an adjustment subtracts $\lambda^{n_r\ \text{Ukrainian}}_{\neg n_r\ \text{Ukrainian}}$ from the original $\lambda^{n_t\ \text{Russian}}_{\neg n_t\ \text{Russian}}$ likelihood ratio. The subtraction operation suppresses the high scores in $\lambda^{n_t\ \text{Russian}}_{\neg n_t\ \text{Russian}}$ in case of a false alarm in Ukrainian, and compensates the low scores in case of a detection miss in Russian. Recall that the error cost for a detection miss is $N-1$ times of the cost for a false alarm (Eq.(7.10) and (7.11)). So the biggest concerns are those Russian speech segments having large scores in $\lambda^{n_r\ \text{Ukrainian}}_{\neg n_r\ \text{Ukrainian}}$, which will incur detection misses of Russian after the subtraction operation in Eq.(7.12). As a result, the prerequisite for a negative α to be optimal is a low false alarm rate in the detector of the related language. In Table 7.3, $P_{\text{FA}}(n_t\text{:Ukrainian}, n_r\text{:Russian})$ is only 3.25%. A subtraction will not incur detection misses of Russian. Similarly, scores of the Dari detector have relatively low false alarm rate in Farsi (14.32%), and it is subtracted from the scores of the Farsi detector.

The second scenario occurs for the detector n_t where false alarm rate is high in the detector of the related language. The optimal α_{n_t,n_r} parameters found by Eq.(7.13) are non-negative. This is because subtraction of scores would incur a significant number of detection misses, which means a high cost $C_{\text{detect}}(n_t)$ contributing to the average error. In the score adjustment of American and

Indian English, the optimal value of α_{n_t,n_r} are found to be around zero. For other detectors, optimal values of α_{n_t,n_r} are positive. Essentially the adjusted score is a weighted sum of scores from n_t and n_r detectors. The two related languages are less differentiated, in return for fewer detection misses $P_{\text{Miss}}(n_t)$, and/or fewer false alarms irrelevant to the related language pairs $P_{\text{FA}}(n_t, n_n | n_n \notin \{n_t \cup n_r\})$.

After score calibration, the *global error terms* of all languages except Russian is reduced. However, the confusion between the pair of related languages is actually increased. Calibration towards a lower global error somehow sacrifices the differentiation between a target language n_t and its related language n_r. An extra experiment is performed, in which only the confusions between n_t and n_r are looked at [111]. Another cost function is defined and search for the optimization parameters is repeated. Irrelevant imposter data which belong to neither n_t nor n_r are removed. In such case, all α_{n_t,n_r} parameters are found to be negative, and the confusions between n_t and n_r are slightly decreased.

Comparison of the results in different systems

In the final experiment, detection target dependent calibration is repeated with two other configurations used in previous experiments in Table 7.1. They are the PPRVSM scores, and the fused scores between PPRVSM and the partial set (II) [**EXPLOIT**], which models boundary n-grams explicitly but with **Intensity basic** attributes removed. Fusion results are recorded in Table 7.4. To make results comparable to former experiments, both *pooled EER* ($C_{\text{p-eer}}$) and *average EER* (C_{eer}) are shown.

Empirical results show that *pooled EER* is often higher than *average EER*. This is because the constant threshold across all target languages cannot handle score mismatch among different detectors. There are actually opinions saying that the use of a common scale for multiple detections of different targets is not desirable [112, 113].

Detection target dependent calibration improves the LID performance in an ubiquitous way. It brings a 13% relative *average EER* reduction and a 22%-25% relative *pooled EER* reduction in different LID fused systems. The improvements by the score-level calibration is orthogonal to feature-level system fusion In other words, the error reduction brought by prosodic features is still noticeable after the applying calibration The lowest error is observed in the phonotactic-prosodic fused system with detection target dependent calibration Average EER (C_{eer}) and pooled EER ($C_{\text{p eer}}$) are 2 67% and 3 30% respectively

Table 7 4 *Average EER and pooled EER before and after target dependent calibration (NIST LRE 2009)*

LID system	Before calibration (with $\lambda^{n_t}_{\neg n_t}$)		After calibration (with $\lambda'^{n_t}_{\neg n_t}$)	
	C_{eer}	$C_{\text{p-eer}}$	C_{eer}	$C_{\text{p eer}}$
PPRVSM	3 56%	4 63%	3 08%	3 61%
PPRVSM + 7 partial sets[♯]	3 18%	4 45%	2 78%	3 33%
PPRVSM + partial set (II) [EXPLOIT][*]	3 08%	4 42%	2 67%	3 30%

[♯] A partial set is formed by excluding attributes in one of the prosodic groups (I) to (VII)

[*] Partial set (II) with Intensity basic attributes excluded, modeling boundary n grams explicitly

Chapter 8

Conclusions

8.1 Summary

In this thesis, the use of prosodic features for LID is validated with three important operations. First, a large set of prosodic attributes are used in LID experiments. Second, an information-theoretic approach is used to analyze and select among different attributes. Third, score-level fusion with a state-of-the-art phonotactic LID system is performed.

The large set of features proposed in this study covers 105 prosodic attributes. This contrasts to many previous studies, where only a small number of prosodic attributes were involved. With the prosodic attribute model (PAM) approach, a super term-document matrix is constructed to model the cross-attribute correlations and long-range sequential information of prosodic tokens in a flexible manner. The PAM-based prosodic LID system is compared with other prosodic LID systems, and is shown to perform the best. Various constructions of term-document matrices are tested with NIST LRE 2009. The test results validate the use of a comprehensive attribute set, and help to find a preferred construction for the super term-document matrix.

There is a large number of prosodic features involved in this study. An information-theoretic approach is used to select among different normalization and regression methods to give a 14-attribute feature set. Also, a refined metric of bin-level mutual information evaluates the prosodic token bigrams in a language-dependent manner, and reveals the prosodic characteristics of specific target languages.

The PAM-based prosodic LID system not only works well as a stand-alone system, but also provides complementary effects to a state-of-the-art phonotactic LID system by score-level fusion. Performance of a phonotactic-prosodic fused system is generally better if the stand-alone prosodic system has fewer errors. In the optimal setting, the inclusion of prosodic features reduces the equal error rate from 3.56% to 3.08% in a closed-set language detection task. Detection target dependent calibration is also carried out and further reduction of errors is observed.

8.2 Contribution of the work

State-of-the-art spoken language identification (LID) systems make use of large-scale acoustic/phonotactic modeling. Prosody features are alternative. Their usage in an LID task has been studied sporadically in the past decade, but investigations were limited to a few specific types of F0 or duration features. Also, there are very few studies that promote prosodic features to large-scale LID tasks. To our knowledge, we are the only participant in NIST Language Recognition Evaluation (LRE) 2009 who makes use of prosodic features in an LID system. In this thesis, we show a comprehensive modeling approach to a large number of prosodic features in large-scale LID tasks. This gives convincing evidence of the effectiveness of prosodic features to distinguish spoken languages.

In this thesis, a couple of machine learning techniques are applied in the course of language detection. An information-theoretic approach is proposed to analyze the effectiveness of different prosodic attributes and to facilitate a feature selection process for dimension reduction. A convex optimization approach is proposed for the backend processing of score fusion and calibration. These approaches are likely to work well in other empirical tasks which may or may not be with prosodic features.

8.3 Limitations and Future work

Most of the LID experiments reported in this thesis are based on the 14-attribute feature set. Compared with the over-complete representation of 105 prosodic attributes, there may be much information which is not adequately modeled. There are attempts to expand the 14-attribute feature set by including various measurements/normalization methods; and n-gram modeling is extended up to 5-gram. These changes significantly increase the problem dimension, but only marginal performance improvements are observed. In the future, some efficient methods to incorporate new information to the PAM-based prosodic LID model are expected to help. For instance, parallel streams of similar attributes can be modeled in a lattice. Feature-level fusion can be done.

Despite having a large number, the 105 prosodic attributes by no means represent the complete prosodic space. As there is not a conventional set of prosodic features, we do not know exactly which prosodic attributes to extract for the LID task. Apart from F0, intensity and duration, other elements, such as timbre, pauses, tempo, etc, are sometimes considered to be elements of prosody. The appropriate modeling of these elements may provide useful information to LID. The interaction between spectral and prosodic features is another important phenomenon to model.

According to [5, 43], the effectiveness of prosodic attributes in LID varies across languages. The bin-level mutual information metric is intended to serve the purpose of deriving language-dependent feature sets for system training. However, results in Section 6.2.5 show that such feature sets only gives marginal improvements. In a follow-up experiment, we added score fusion and calibration modules in the backend. The LID error with language-specific feature set was 25% relatively lower than using the language-independent set (with fusion and calibration) [114]. Further studies can be done along this line.

A numerical adjustment to score is introduced in Chapter 7 to improve LID accuracy. While this operation significantly reduces the pooled error, the distinguishability among some similar languages is actually sacrificed (Table 7.3). That reminds us of different LID scenarios where different handling methods apply. Towards developing a real-world LID application, considerations to its evaluation metric are important.

Bibliography

[1] C. Becchetti and L. P. Ricotti, "Speech signal analysis - theory," in *Speech recognition - Theory and C++ Implementation*, Rome, Italy, 1999, pp. 122–143.

[2] 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, Jan. 1996.

[3] Y. K. Muthusamy, E. Barnard, and R. A. Cole, "Reviewing automatic language identification," *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 33–41, Oct. 1994.

[4] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 271–284, Jan. 2007.

[5] J. Navrátil, "Automatic language identification," in *Multilingual Speech Processing*, T. Schultz and K. Kirchhoff, Eds. Academic Press, 2006, pp. 233–272.

[6] Y. K. Muthusamy and A. L. Spitz, "Automatic language identification," in *Survey of the state of the art in human language technology*, A. Zaenen, Ed., 1996.

[7] K. Atkinson, "Language identification from nonsegmental cues," *Journal of the Acoustical Society of America*, vol. 44, no. 1, p. 378, Jul. 1968.

[8] T. Hanley, J. Snidecor, and R. Ringel, "Some acoustic differences among languages," *Phonetica*, vol. 14, pp. 97–107, 1966.

[9] *The 2009 NIST Language Recognition Evaluation Results.* [Online]. Available: http://www.itl.nist.gov/iad/mig//tests/lre/2009/lre09_eval_results/index.html

[10] J.-L. Rouas, J. Farinas, F. Pellegrino, and R. André-Obrecht, "Rhythmic unit extraction and modelling for automatic language identification," *Speech Communication*, vol. 47, no. 4, pp. 436–456, Dec. 2005.

[11] F. Ramus and J. Mehler, "Language identification with suprasegmental cues: A study based on speech resynthesis," *Journal of the Acoustical Society of America*, vol. 105, no. 1, pp. 512–521, Jan. 1999.

[12] J.-L. Rouas, "Automatic prosodic variations modeling for language and dialect discrimination," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 6, pp. 1904–1911, Aug. 2007.

[13] *The 2007 NIST Language Recognition Evaluation Plan (LRE07).* [Online]. Available: http://www.itl.nist.gov/iad/mig/tests/lre/2007/LRE07EvalPlan-v8b.pdf

[14] *The 2009 NIST Language Recognition Evaluation Plan (LRE09).* [Online]. Available: http://www.itl.nist.gov/iad/mig//tests/lre/2009/LRE09_EvalPlan_v6.pdf

[15] D. Cimarusti and R. B. Ives, "Development of an automatic identification system of spoken languages: Phase I," in *Proceedings of ICASSP*, 1982, pp. 1661–1663.

[16] M. Sugiyama, "Automatic language recognition using acoustic features," in *Proceedings of ICASSP*, 1991, pp. 813–816.

[17] R. D. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, "An efficient auditory filterbank based on the gammatone function," in *APU Rep. 2341.* Cambridge, U.K.: Appl. Psychol. Unit,Cambridge Univ., 1988.

[18] H. Li, B. Ma, K. A. Lee, C. You, R. Tong, D. Zhu, H. Sun, K. C. Sim, C. C. Leung, C.-L. Huang, and I. Karkkainen, *IIR system description for the 2009 NIST Language Recognition Evaluation*, 2009.

[19] N. Brümmer, L. Burget, O. Glembek, V. Hubeika, Z. Jančík, M. Karafiát, P. Matějka, T. Mikolov, O. Plchot, and A. Strasheim, *BUT-AGNITIO system description for NIST Language Recognition Evaluation 2009*, 2009.

[20] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, Apr. 1990.

[21] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, Apr. 1975.

[22] R. Heab-Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition," in *Proceedings of ICASSP*, vol. 1, 1992, pp. 13–16.

[23] L. Lee and R. C. Rose, "A frequency warping approach to speaker normalization," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 1, pp. 49–60, Jan. 1998.

[24] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, Oct. 1994.

[25] P. A. Torres-Carrasquillo, E. Singer, W. Campbell, T. Gleason, A. McCree, D. A. Reynolds, F. Richardson, W. Shen, and D. Sturim, "The MITLL NIST LRE 2007 language recognition system," in *Proceedings of Interspeech*, 2008, pp. 719–722.

[26] M. Ferràs, C.-C. Leung, C. Barras, and J.-L. Gauvain, "Constrained MLLR for speaker recognition," in *Proceedings of ICASSP*, vol. IV, 2007, pp. 53–56.

[27] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Factor analysis simplified," in *Proceedings of ICASSP*, vol. I, 2005, pp. 637–640.

[28] P. Kenny, G. Boulianne, P. Oueleet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1435–1447, May. 2007.

[29] O. Glembek, P. Matějka, L. Burget, and T. Mikolov, "Advances in phonotactic language recognition," in *Proceedings of Interspeech*, 2008, pp. 743–746.

[30] J. A. Bilmes, "A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," *Technical Report ICSI-TR-97-02*, Apr. 1998.

[31] L. Burget, P. Matějka, and J. Černocký, "Discriminative training techniques for acoustic language identification," in *Proceedings of ICASSP*, vol. I, 2006, pp. 209–212.

[32] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," vol. 5, no. 3, pp. 257–265, May. 1997.

[33] D. Povey and P. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proceedings of ICASSP*, 2002, pp. 105–108.

[34] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proceedings of ICASSP*, vol. I, 2005, pp. 961–964.

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

[36] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, “Acoustic language identification using fast discriminative training,” in *Proceedings of Interspeech*, 2007, pp. 346–349.

[37] W. M. Campbell, “A covariance kernel for SVM language recognition,” in *Proceedings of ICASSP*, 2008, pp. 4141–4144.

[38] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. Reynolds, and J. J. R. Deller, “Approaches to language identification using Gaussian mixture models and shifted delta cepstral features,” in *Proceedings of ICSLP*, 2002.

[39] H. Li and B. Ma, “A phonotactic language model for spoken language identification,” in *ACL ’05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005, pp. 515–522.

[40] W. M. Campbell, F. Richardson, and D. A. Reynolds, “Language recognition with word lattices and support vector machines,” in *Proceedings of ICASSP*, 2007, pp. 989–992.

[41] J. Gauvain, A. Messaoudi, and H. Schewenk, “Language recognition using phone lattices,” in *Proceedings of ICSLP*, 2004, pp. 1215–1218.

[42] Y. K. Muthusamy, N. Jain, and R. A. Cole, “Perceptual benchmarks for automatic language identification,” in *Proc. ICASSP*, vol. I, 1994, pp. 333–336.

[43] A. E. Thymé-Gobbel and S. E. Hutchins, “On using prosodic cues in automatic language identification,” in *Proceedings of ICSLP*, 1996, pp. 1768–1771.

[44] S. Eady, “Differences in the F_0 patterns of speech: Tone language versus stress language,” *Language and speech*, vol. 25, no. 1, pp. 29–42, Jan.-Mar. 1982.

[45] C.-Y. Lin and H.-C. Wang, "Language identification using pitch contour information in the ergodic Markov model," in *Proceedings of ICASSP*, 2006, pp. 193–196.

[46] L. Mary and B. Yegnanarayana, "Extraction and representation of prosodic features for language and speaker recognition," *Speech Communication*, vol. 50, no. 10, pp. 782–796, Oct. 2008.

[47] L. M. Hyman, "Word-prosodic typology," *Phonology*, vol. 23, no. 2, pp. 225–257, Aug. 2006.

[48] I. Lehiste, *Suprasegmentals.* Cambridge, MA, US: MIT Press, 1970.

[49] M. Liberman, "On stress and linguistic rhythm," *Linguistic Inquiry*, vol. 8, no. 2, pp. 249–336, 1977.

[50] L. J. Downing, "What African languages tell us about accent typology," *ZAS Papers in Linguistics*, vol. 37, pp. 101–136, 2004.

[51] M. Yip, "Tones in east asian languages," in *The Handbook of Phonological Theory*, J. Goldsmith, Ed. Cambridge, MA: Blackwell, 1995, pp. 476–494.

[52] Y. Xu, "Speech melody as articulatorily implemented communicative functions," *Speech Communication*, vol. 46, no. 3-4, pp. 220–251, Jul. 2005.

[53] S. Inkelas and D. Zec, "Serbo-croatian pitch accent: The interaction of tone, stress, and intonation," *Language*, vol. 64, no. 2, pp. 227–248, 1988.

[54] H. R. Pfitzinger, S. Burger, and S. Heid, "Syllable detection in read and spontaneous speech," in *Proceedings of ICSLP*, 1996, pp. 1261–1264.

[55] F. Biadsy and J. Hirschberg, "Using prosody and phonotactics in Arabic dialect identification," in *Proceedings of Interspeech*, 2009, pp. 208–211.

[56] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke, "Modeling prosodic feature sequences for speaker recognition," *Speech Communication*, vol. 46, no. 3-4, pp. 455–472, Jul. 2005.

[57] P. Yin, E. Ambikairajah, and F. Chen, "Voiced/unvoiced pattern-based duration modeling for language identification," in *Proceedings of ICASSP*, 2009, pp. 4341–4344.

[58] E. Timoshenko and H. Höge, "Using speech rhythm for acoustic language identification," in *Proceedings of Interspeech*, 2007, pp. 182–185.

[59] G. Kochanski, A. Loukina, E. Keane, C. Shih, and B. Rosner, "Long-range prosody prediction and rhythm," in *Proceedings of Speech Prosody*, 2010.

[60] C.-Y. Lin and H.-C. Wang, "Language identification using pitch contour information," in *Proceedings of ICASSP*, 2005, pp. 601–604.

[61] A. G. Adami and H. Hermansky, "Segmentation of speech for speaker and language recognition," in *Proceedings of Eurospeech*, 2003, pp. 841–844.

[62] J.-L. Rouas, J. Farinas, F. Pellegrino, and R. André-Obrecht, "Modeling prosody for language identification on read and spontaneous speech," in *Proceedings of ICASSP*, 2003, pp. 40–43.

[63] M. Piat, D. Fohr, and I. Illina, "Foreign accent identification based on prosodic parameters," in *Proceedings of Interspeech*, 2008, pp. 759–762.

[64] G.-A. Levow, "Unsupervised and semi-supervised learning of tone and pitch accent," in *Proceedings of HLT-NAACL*, 2006, pp. 224–231.

[65] G. Peng and W. S.-Y. Wang, "Tone recognition of continuous cantonese speech based on support vector machines," *Speech Communication*, vol. 45, no. 1, pp. 49–62, Jan. 2005.

[66] T. J. Hazen and V. W. Zue, "Segment-based automatic language identification," *Journal of the Acoustical Society of America*, vol. 101, no. 4, pp. 2324–2331, Apr. 1997.

[67] S. Kajarekar, L. Ferrer, K. Sönmez, J. Zheng, E. Shriberg, and A. Stolcke, "Modeling NERFs for speaker recognition," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2004, pp. 51–56.

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

[69] S. Abe, *Support Vector Machines for Pattern Classification (Advances in Pattern Recognition).* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[70] F. S. Richardson, W. M. Campbell, and P. A. Torres-Carrasquillo, "Discriminative n-gram selection for dialect recognition," in *Proceedings of Interspeech*, 2009, pp. 192–195.

[71] L.-F. Zhai, M. hung Siu, X. Yang, and H. Gish, "Discriminatively trained language models using support vector machines for language identification," in *Proceedings of IEEE Odyssey: The Speaker and Language Recognition Workshop*, 2006, pp. 1 –6.

[72] H. Suo, M. Li, P. Lu, and Y. Yan, "Using SVM as back-end classifier for language identification," *EURASIP J. Audio Speech Music Process.*, vol. 2008, pp. 1–6, 2008.

[73] E. Shriberg, L. Ferrer, and S. Kajarekar, "SVM modeling of SNERF-grams for speaker recognition," in *Proceedings of ICSLP*, 2004, pp. 1409–1412.

[74] X. Yang, L.-F. Zhai, M. Siu, and H. Gish, "Improved language identification using support vector machines for language modeling," in *Proceedings of Interspeech*, 2006, pp. 417–420.

[75] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[76] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[77] T. Joachims, "Making large-scale SVM learning practical," in *Advances in kernel methods - support vector learning*, B. Schölkopt, C. Burges, and A. Smola, Eds. Cambridge, MA, US: MIT Press, 1999.

[78] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using second order information for training support vector machines," *Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.

[79] *Implementation of Support Vector Machines in C.* [Online]. Available: http://svmlight.joachims.org

[80] W. Weigel, "Silbenorientierte erkennung fließender Sprache mittels diskreter stochastischer Modellierung," Ph.D. dissertation, TU München, 1990.

[81] P. Mermelstein, "Automatic segmentation of speech into syllabic units," *Journal of the Acoustical Society of America*, vol. 58, no. 4, pp. 880–883, Oct. 1975.

[82] A. W. Howitt, "Vowel landmark detection," in *Proceedings of Eurospeech*, 1999, pp. 2777–2780.

[83] F. Pellegrino and R. Andre-Obrecht, "Automatic language identification: an alternative approach to phonetic modeling," *Signal Processing*, vol. 80, no. 7, pp. 1231–1244, 2000.

[84] Entropic, *ESPS Version 5.0 Programs Manual*, Washington, D.C., US, 1993.

[85] T. Lee and Y. Qian, "Tone modeling for speech recognition," in *Advances in Chinese Spoken Language Processing*, C.-H. L. et al., Ed. Singapore: World Scientific Publishing, 2007.

[86] E. Shriberg and L. Ferrer, "A text-constrained prosodic system for speaker verification," in *Proceedings of Interspeech*, 2007, pp. 1226–1229.

[87] *The 1996 NIST Language Recognition Evaluation Plan (LRE96).* [Online]. Available: http://www.itl.nist.gov/iad/mig/tests/lre/1996/LRE96EvalPlan.pdf

[88] *The 2003 NIST Language Recognition Evaluation Plan (LRE03).* [Online]. Available: http://www.itl.nist.gov/iad/mig/tests/lre/2003/LRE03EvalPlan-v1.pdf

[89] Y. K. Muthusamy, R. Cole, and B. Oshika, "The OGI multilanguage telephone speech corpus," in *Proceedings of ICSLP*, 1992, pp. 895–898.

[90] S. Gao, B. Ma, H. Li, and C.-H. Lee, "A text-categorization approach to spoken language identification," in *Proceedings of Interspeech*, 2005, pp. 2837–2840.

[91] G. Salton, *The SMART Retrieval System.* NY, USA: Prentice Hall, 1971.

[92] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[93] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, Sep. 1990.

[94] J. Chu-Carroll and B. Carpenter, "Vector-based natural language call routing," *Computational Linguistics*, vol. 25, no. 3, pp. 361–388, 1999.

[95] E. Shriberg, "Higher-level features in speaker recognition," in *Speaker Classification I*, ser. Lecture Notes in Computer Science, C. Müller, Ed. Springer Berlin / Heideberg, 2007, vol. 4343, pp. 241–259.

[96] J. Goodman, "A bit of progress in language modeling," *Computer Speech and Language*, vol. 15, no. 4, pp. 403–434, Oct. 2001.

[97] R. Blasig, "Combination of words and word categories in varigram histories," in *Proceedings of ICASSP*, 1999, pp. 529–532.

[98] S. Martin, J. Liermann, and H. Ney, "Algorithms for bigram and trigram word clustering," *Speech Communication*, vol. 24, no. 1, pp. 19–37, Apr. 1998.

[99] A. Stolcke, E. Shriberg, D. Hakkani-Tür, and G. Tür, "Modeling the prosody of hidden events for improved word recognition," in *Proceedings of Eurospeech*, 1999, pp. 311–314.

[100] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994.

[101] D. A. van Leeuwen and N. Brümmer, "Building language detectors using small amounts of training data," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2008.

[102] N. Brümmer and J. du Preez, "Application-independent evaluation of speaker detection," *Computer, Speech and Language*, vol. 20, no. 2-3, pp. 230–275, 2006.

[103] A. Cutler and T. Otake, "Pitch accent in spoken-word recognition in Japanese," *J. Acoust. Soc. Am.*, vol. 105, no. 3, pp. 1877–1888, Mar. 1999.

[104] E. Keane, "Prominence in Tamil," *J. International Phonetic Association*, vol. 36, pp. 1–20, May. 2006.

[105] M. A. Zissman and K. M. Berkling, "Automatic language identification," *Speech Communication*, vol. 35, no. 1-2, pp. 115–124, Aug. 2001.

[106] R. Tong, B. Ma, D. Zhu, H. Li, and E. S. Chng, "Integrating acoustic, prosodic and phonotactic features for spoken language identification," in *Proceedings of ICASSP*, 2006, pp. 205–208.

[107] Z. Chair and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-22, pp. 98–101, Jan. 1986.

[108] C. C. Leung, R. Tong, B. Ma, and H. Li, "A lattice-based phonotactic language recognition system with CMLLR adaptation and its implementation issues," in *Proceedings in International Conference on Asian Language Processing*, 2009, pp. 285–288.

[109] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, UK: Cambridge University Press, 2004.

[110] M. Grang and S. Boyd, *CVX: Matlab software for disciplined convex programming*, Jun. 2009. [Online]. Available: http://standard.edu/boyd/cvx

[111] R. W. M. Ng, C.-C. Leung, T. Lee, B. Ma, and H. Li, "Detection target dependent score calibration for language recognition," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2010, pp. 91–96.

[112] N. Brümmer and D. A. van Leeuwen, "On calibration of language recognition scores," in *Proceedings of IEEE Odyssey: The Speaker and Language Recognition Workshop*, 2006, pp. 1–8.

[113] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proceedings of Interspeech*, 2007, pp. 1895–1898.

[114] R. W. M. Ng, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Spoken language identification with prosodic features," 2011, submitted to *IEEE Transactions on Audio, Speech and Language Processing.*

Glossary

application-independent calibration 93

average cost performance 71

average equal error rate (EER) 75

bias removal 37

boundary bigram 49

boundary 49

calibration 70

cepstral mean substraction (CMS) 7

closed-set detection 71

contours 16

count vector 46

detection error tradeoff (DET) curve 72

equal error rate (EER) 72

frame-based measurements 35

Gaussian backend 70

Gaussian supervector 10

generalized linear discriminant sequence kernel (GLDS kernel) 10

global error terms 103

inventory size/cardinality 44

inventory 44

margin 19

maximum mutual information 9

multi-class likelihood score vector 69

mutual information 58

normalization 37

open-set detection 71

pair contour 34

parallel phone recognition followed by language modeling (PPRLM) 11

peak-picking syllabification (PPS) 25

pooled equal error rate (EER) 97

prosodic attribute model (PAM) 51

prosodic attribute 31

prosodic phone 52

prosodic token 44

pseudosyllable 13

relative spectra (RASTA) filtering 7

segment contour 33

shifted delta cepstra (SDC) 10

single language detector 67

skipping *n*-gram 48

soft margin 20

super term-document matrix 50

support vector machine 18

syllable-based measurements 36

syllable-based non uniform extraction region features (SNERF) ... 17

syntagmatic 18

term-document matrix 46

total erroneous deviation 99

triplet contour 34

utterance contour 34

z-normalization 37