Acoustic modeling for speech recognition under limited training data conditions

Do, Van Hai

2015

Do, V. H. (2015). Acoustic modeling for speech recognition under limited training data conditions. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65409>

<https://doi.org/10.32657/10356/65409>

ACOUSTIC MODELING FOR SPEECH RECOGNITION UNDER LIMITED TRAINING DATA CONDITIONS

A thesis submitted to the School of Computer Engineering
in partial fulfillment of the requirement for
the degree of Doctor of Philosophy

by

DO VAN HAI

August 26, 2015

Abstract

The development of a speech recognition system requires at least three resources: a large labeled speech corpus to build the acoustic model, a pronunciation lexicon to map words to phone sequences, and a large text corpus to build the language model. For many languages such as dialects or minority languages, these resources are limited or even unavailable - we label these languages as under-resourced. In this thesis, the focus is to develop reliable acoustic models for under-resourced languages. The following three works have been proposed.

In the first work, reliable acoustic models are built by transferring acoustic information from well-resourced languages (source) to under-resourced languages (target). Specifically, the phone models of the source language are reused to form the phone models of the target language. This is motivated by the fact that all human languages share a similar acoustic space, and hence some acoustic units e.g. phones, of two languages may have high correspondence and therefore allows the mapping of phones between languages. Unlike previous studies which examined only context-independent phone mapping, the thesis extends the studies to use context-dependent triphone states as the units to achieve higher acoustic resolution. In addition, linear and nonlinear mapping models with different training algorithms are also investigated. The results show that the nonlinear mapping with discriminative training criterion achieves the best performance in the proposed work.

In the second work, rather than increasing the mapping resolution, the focus is to improve the quality of the cross-lingual feature used for mapping. Two approaches based on deep neural networks (DNNs) are examined. First, DNNs are used as the source language acoustic model to generate posterior features for phone mapping. Second, DNNs are used to replace multilayer perceptrons (MLPs) to realize the phone mapping.

Experimental results show that better phone posteriors generated from the source DNNs result in a significant improvement in cross-lingual phone mapping, while deep structures for phone mapping are only useful when sufficient target language training data are available.

The third work focuses on building a robust acoustic model using the exemplar-based modeling technique. Exemplar-based model is non-parametric and uses the training samples directly during recognition without training model parameters. This study uses a specific exemplar-based model, called kernel density, to estimate the likelihood of target language triphone states. To improve performance for under-resourced languages, cross-lingual bottleneck feature is used. In the exemplar-based technique, the major design consideration is the choice of distance function used to measure the similarity of a test sample and a training sample. This work proposed a Mahalanobis distance based metric optimized by minimizing the classification error rate on the training data. Results show that the proposed distance produces better results than the Euclidean distance. In addition, a discriminative score tuning network, using the same principle of minimizing training classification error, is also proposed.

Acknowledgments

I would like to express my sincere thanks and appreciation to my supervisor, Dr. Chng Eng Siong (NTU), and co-supervisor, Dr. Li Haizhou (I²R) for their invaluable guidance, support and suggestions. Their encouragement also helps me to overcome the difficulties encountered in my research. My thanks also go to Dr. Xiao Xiong (NTU) for his help, discussions during my PhD study time.

My thanks also go to my colleagues in the speech group of the International Computer Science Institute (ICSI) including Prof. Nelson Morgan, Dr. Steven Wegmann, Dr. Adam Janin, Arlo Faria for their generous help and fruitful discussions during my internship at ICSI.

I also want to thank my colleagues in the speech group in NTU, for their help. I am very comfortable to collaborate with my team mates Guangpu, Ha, Haihua, Hy, Nga, Steven, Thanh, Tung, Tze Yuang and Zhizheng. My life is much more colorful and comfortable with their friendship.

Last but not least, I would like to thank my family in Vietnam, for their constant love and encouragement.

Contents

Abstract	i
Acknowledgments	iii
List of Publications	vii
List of Figures	ix
List of Tables	xii
List of Abbreviations	xv
1 Introduction	1
1.1 Contributions	3
1.2 Thesis Outline	5
2 Fundamentals and Previous Works	7
2.1 Typical Speech Recognition System	7
2.1.1 Feature extraction	8
2.1.2 Acoustic model	8
2.1.3 Language model and word lexicon	11
2.1.4 Decoding	11
2.2 Acoustic Modeling for Under-resourced Languages	12
2.2.1 Universal phone set method	14
2.2.2 Cross-lingual subspace GMM (SGMM)	19
2.2.3 Source language models act as a feature extractor	22
2.3 Summary	33

3	Cross-lingual Phone Mapping	35
3.1	Tasks and Databases	36
3.2	Cross-lingual Linear Acoustic Model Combination	37
3.2.1	One-to-one phone mapping	37
3.2.2	Many-to-one phone mapping	39
3.3	Cross-lingual Nonlinear Context-dependent Phone Mapping	44
3.3.1	Context-independent phone mapping	45
3.3.2	Context-dependent phone mapping	46
3.3.3	Combination of different input types for phone mapping	49
3.3.4	Experiments	50
3.4	Conclusion	65
4	Deep Neural Networks for Cross-lingual Phone Mapping Framework	67
4.1	Deep Neural Network Introduction	68
4.1.1	Deep architectures	68
4.1.2	Restricted Boltzmann machines	69
4.1.3	Deep neural networks for speech recognition	71
4.1.4	Experiments of using DNNs for monolingual speech recognition	73
4.1.5	Conclusion	78
4.2	Deep Neural Networks for Cross-lingual Phone Mapping	79
4.2.1	Three setups using DNNs for cross-lingual phone mapping	80
4.2.2	Experiments	83
4.3	Conclusion	88
5	Exemplar-based Acoustic Models for Limited Training Data Conditions	90
5.1	Introduction	91
5.2	Kernel Density Model for Acoustic Modeling	93
5.3	Distance Metric Learning	95
5.4	Discriminative Score Tuning	97
5.5	Experiments	98
5.5.1	Experimental procedures	98

5.5.2	Using MFCC and cross-lingual bottleneck feature as the acoustic feature	99
5.5.3	Distance metric learning for kernel density model	101
5.5.4	Discriminative score tuning	103
5.6	Conclusion	107
6	Conclusion and Future Works	108
6.1	Contributions	108
6.1.1	Context-dependent phone mapping	108
6.1.2	Deep neural networks for phone mapping	109
6.1.3	Exemplar-based acoustic model	110
6.2	Future Directions	110
A	Appendix	112
A.1	Derivation for Distance Metric Learning	112
	References	116

List of Publications

Journals

- (i) **Van Hai Do**, Xiong Xiao, Eng Siong Chng, and Haizhou Li, “Context-dependent Phone Mapping for Acoustic Modeling of Under-resourced Languages,” *International Journal of Asian Language Processing*, vol. 23, no. 1, pp. 21-33, 2015.
- (ii) **Van Hai Do**, Xiong Xiao, Eng Siong Chng, and Haizhou Li, “Cross-lingual Phone Mapping for Large Vocabulary Speech Recognition of Under-resourced Languages,” *IEICE Transactions on Information and Systems*, vol. E97-D, no. 2, pp. 285-295, 2014.

Conferences

- (iii) **Van Hai Do**, Xiong Xiao, Eng Siong Chng, and Haizhou Li, “Kernel Density-based Acoustic Model with Cross-lingual Bottleneck Features for Resource Limited LVCSR,” in *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 6-10, 2014.
- (iv) **Van Hai Do**, Xiong Xiao, Eng Siong Chng, and Haizhou Li, “Context-dependent Phone Mapping for LVCSR of Under-resourced Languages,” in *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 500-504, 2013.
- (v) **Van Hai Do**, Xiong Xiao, Eng Siong Chng, and Haizhou Li, “Context Dependant Phone Mapping for Cross-Lingual Acoustic Modeling,” in *Proceedings of International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 16-20, 2012.

- (vi) **Van Hai Do**, Xiong Xiao, Eng Siong Chng, and Haizhou Li, “A Phone Mapping Technique for Acoustic Modeling of Under-resourced Languages,” in *Proceedings of International Conference on Asian Language Processing (IALP)*, pp. 233-236, 2012.
- (vii) **Van Hai Do**, Xiong Xiao, and Eng Siong Chng, “Comparison and Combination of Multilayer Perceptrons and Deep Belief Networks in Hybrid Automatic Speech Recognition Systems”, in *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2011.
- (viii) **Van Hai Do**, Xiong Xiao, Ville Hautamaki, and Eng Siong Chng, “Speech Attribute Recognition using Context-Dependent Modeling”, in *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2011.
- (ix) Haihua Xu, **Van Hai Do**, Xiong Xiao, and Eng Siong Chng, “A Comparative Study of BNF and DNN Multilingual Training on Cross-lingual Low-resource Speech Recognition,” in *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.
- (x) Mirco Ravanelli, **Van Hai Do**, and Adam Janin “TANDEM-Bottleneck Feature Combination using Hierarchical Deep Neural Networks,” in *Proceedings of International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 113-117, 2014.
- (xi) Korbinian Riedhammer, **Van Hai Do**, and Jim Hieronymus, “A Study on LVCSR and Keyword Search for Tagalog,” in *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2529-2533, 2013.

List of Figures

2.1	Block diagram of a typical speech recognition system.	8
2.2	A left-to-right HMM model with three true states.	9
2.3	Phonetic-acoustic modeling [1].	10
2.4	An example of the universal phone set approach.	13
2.5	Illustration of the cross-lingual subspace Gaussian mixture model (SGMM) approach.	14
2.6	Source acoustic models act as a feature extractor.	14
2.7	Phones of Spanish and Japanese are expressed in the IPA format [2]. . .	15
2.8	Cross-lingual tandem approach.	24
2.9	Automatic speech attribute transcription (ASAT) framework for cross-lingual speech recognition.	25
2.10	Probabilistic sequence-based phone mapping model.	30
3.1	Differences between feature distributions of three HMM states for a same IPA phone across three languages [3].	40
3.2	Combination weights for the proposed many-to-one phone mapping. . . .	42
3.3	Combination weights for phone “ <i>sh</i> ” and “ <i>th</i> ” in English.	43
3.4	Posterior-based phone mapping. N_S and N_T are the numbers of phones in the source and target languages, respectively.	45
3.5	A diagram of the training process for the context-dependent cross-lingual phone-mapping.	48
3.6	A diagram of the decoding process for the context-dependent cross-lingual phone mapping.	49

3.7	Feature combination in the cross-lingual phone mapping where N_{S1}, N_{S2} are the numbers of tied-states in the source language acoustic model 1 and 2, respectively.	50
3.8	Probability combination in the cross-lingual phone mapping.	50
3.9	Cross-lingual tandem system for Malay (source language) and English (target language).	55
3.10	The WER (%) of different acoustic models with different amounts of target training data.	61
3.11	The WER (%) of individual and combined phone mapping models. . . .	62
3.12	The WER (%) of the phone mapping model for two source languages with three different amounts of target training data.	64
4.1	A DNN (b) is composed by a stack of RBMs (a).	69
4.2	Four steps in training a DNN for speech recognition.	73
4.3	Phone error rate (PER) on the training and test sets of the HMM/GMM model with different model complexities.	75
4.4	Performance of the HMM/DNN acoustic model for the two initialization schemes and the combined models with different numbers of layers. . . .	76
4.5	Comparison of HMM/MLP and HMM/DNN on an LVCSR task for different training data sizes.	78
4.6	Illustration of using DNN as the source language acoustic model to generate cross-lingual posterior feature for phone mapping (<i>Setup A</i>).	81
4.7	Illustration of using DNN to extract cross-lingual bottleneck feature for target language acoustic models (<i>Setup B</i>).	81
4.8	Illustration of using DNNs for both the source acoustic model and the phone mapping (<i>Setup C</i>).	82
4.9	WER (%) of the cross-lingual phone mapping using deep source acoustic model versus the phone mapping using shallow source acoustic model and the monolingual model.	86
4.10	WER (%) of the two cross-lingual models using bottleneck feature generated by the deep source bottleneck network (<i>Setup B</i>). The last column shows the WER given by the phone mapping model in <i>Setup A</i> for comparison.	86

4.11	Comparison of the cross-lingual phone mapping using shallow and deep structures for phone mapping.	88
5.1	Kernel density estimation for acoustic modeling of an LVCSR system. . .	94
5.2	The proposed discriminative score tuning.	97
5.3	Illustration of the linear feature transformation matrices. BN stands for cross-lingual bottleneck features, DML stands for distance metric learning. The MFCC feature vectors are ordered as $[c1,...,c12,c0]$, and their delta and acceleration versions.	102
5.4	Frame error rate (%) obtained by the kernel density model with two initialization schemes for transformation $\mathbf{Q}$	104
5.5	The weight matrix of the 2-layer neural network score tuning.	105
5.6	Performance in WER(%) of the proposed kernel density model using different score tuning neural network architectures.	106

List of Tables

2.1	Performance and model complexity of GMM and SGMM models in a monolingual task [4].	21
2.2	Different cross-lingual methods using the source acoustic models as a feature extractor.	23
2.3	Speech attribute subsets [5].	25
3.1	One-to-one phone mapping from Malay to English generated by the data-driven method using 7 minutes of English training data.	38
3.2	Word error rate (WER) (%) of the one-to-one phone mapping for cross-lingual acoustic models.	39
3.3	Word error rate (WER) (%) of many-to-one phone mapping for cross-lingual acoustic models with 7 minutes of target training data.	43
3.4	Word error rate (WER) (%) of the monolingual monophone and triphone baseline HMM/GMM models for 16 minutes of training data with different model complexities.	54
3.5	The WER (%) of different monolingual and cross-lingual acoustic models with 16 minutes of English training data.	56
3.6	The WER (%) of different mapping architectures with 16 minutes of target English training data (the target acoustic model is triphone). Number in (.) in the first and the last column indicates the number of inputs and number of hidden units, respectively. Number in (.) of the third column represents relative improvement over the corresponding 2-layer NN in the second column.	59

5.1	WER (%) obtained by various models at four different training data sizes. Row 1-6 are results obtained by using MFCC feature. Row 7-11 show results obtained by using cross-lingual bottleneck feature. KD stands for kernel density used for acoustic modeling.	100
5.2	WER (%) obtained by the kernel density model using MFCC feature with two initialization schemes for transformation $\mathbf{Q}$	104

List of Abbreviations

ASAT	Automatic Speech Attribute Transcription system
ASR	Automatic Speech Recognition
BN-DNN	Deep Neural Network to generate cross-lingual BottleNeck feature
DML	Distance Metric Learning
DNN	Deep Neural Network
EM	Expectation Maximization
FER	Frame Error Rate
GMM	Gaussian Mixture Model
HLDA	Heteroscedastic Linear Discriminant Analysis
HMM	Hidden Markov Model
HMM/DNN	Hidden Markov Model / Deep Neural Network
HMM/GMM	Hidden Markov Model / Gaussian Mixture Model
HMM/MLP	Hidden Markov Model / Multi-Layer Perceptron
HMM/KD	Hidden Markov Model / Kernel Density Model
IPA	International Phonetic Alphabet
KD	Kernel Density Model
KL-HMM	Kullback-Leibler based Hidden Markov Model
k-NN	k-Nearest Neighbors
LDA	Linear Discriminant Analysis
LVCSR	Large-Vocabulary Continuous Speech Recognition
MFCC	Mel Frequency Cepstral Coefficients
MAP	Maximum A Posteriori
MLE	Maximum Likelihood Estimation
MLLR	Maximum Likelihood Linear Regression
MLP	Multi-Layer Perceptron
MMI	Maximum Mutual Information
MVN	Mean and Variance Normalization
NN	Neural Network
PCA	Principal Component Analysis
PER	Phone Error Rate
POS-DNN	Deep Neural Network to generate cross-lingual POSterior feature
RBM	Restricted Boltzmann Machine
SA	Speech Attribute
SGMM	Subspace Gaussian Mixture Model
WER	Word Error Rate
WSJ	Wall Street Journal ^{xv} speech corpus

Chapter 1

Introduction

Speech is the most common and effective form of human communication. Much research in the last few decades has focused on improving speech recognition systems to robustly convert speech into text automatically [6]. Unfortunately, speech researchers have only focused on dozens out of the thousands of spoken languages in the world [7]. To build a large-vocabulary continuous speech recognition (LVCSR) system, at least three resources are required: a large labeled speech corpus to build the acoustic model, a pronunciation lexicon to map words to phone sequences, and a large text corpus to build the language model. For many languages such as dialects or minority languages, these resources are limited or even unavailable, they are called under-resourced languages for LVCSR research. The scope of the thesis is to examine the development of reliable acoustic models for under-resourced languages.

One major obstacle to build an acoustic model for a new language is that it is very expensive to acquire a large amount of labeled speech data. Usually tens to hundreds of hours of training data are required to build a reasonable acoustic model for an LVCSR system, while commercial systems normally use thousands of hours of training data. This large resource requirement limits the development of a full fledged acoustic model for under-resourced languages.

The above challenge motivates researchers to examine cross-lingual techniques to transfer knowledge from acoustic models of well-resourced (source) languages to under-resourced (target) languages. Various methods have been proposed for cross-lingual acoustic modeling which can be classified into three main categories.

The first category is based on an universal phone set [7–10] that is generated by merging phone sets of different languages according to the international phonetic alphabet (IPA) scheme. A multilingual acoustic model can therefore be trained for all languages using the common phone set. An initial acoustic model for a new target language can be obtained by mapping from the multilingual acoustic model. To improve performance on the target language, this initial acoustic model is refined using adaptation data of the target language. One disadvantage of this approach is that phones in different languages can be similar, they are unlikely to be identical. Hence, it can be hard to merge different phone sets appropriately.

In the second category, the idea is to create an acoustic model that can be effectively broken down into two parts in which the major part captures language-independent statistics and the other part captures language specific statistics. For cross-lingual acoustic modeling, the language-independent part of a well-trained acoustic model of the source language is borrowed by the acoustic model of the target language to reduce the target language training data requirement. A typical example in this approach is the cross-lingual subspace Gaussian mixture models (SGMMs) [4, 11, 12]. In SGMMs, the model parameters are separated into the subspace parameters that are language-independent, and phone state specific parameters that are language-dependent. With such a model, the well-trained subspace parameters from the source language are captured and speech data of the target language are applied to adapt the target language phone state specific parameters. As those parameters only account for a small proportion of the overall model, they could be reliably trained with a small amount of training data. The disadvantage of this approach is that it requires the source and target models to have the similar structures. This limits the flexibility of the approach for different cross-lingual tasks.

In the third category, which is also the most popular approach, the source acoustic model acts as a feature extractor to generate cross-lingual features such as source language phone posteriors for the target language speech data. As these features are higher-level features as compared to conventional features such as MFCCs, they enable the use of simpler models trained with a small amount of training data to model the target acoustic space. In this category, various cross-lingual methods have been proposed using various types of cross-lingual features and target acoustic models as detailed below.

- *Cross-lingual tandem*: phone posteriors [13–15] generated by source language multilayer perceptron neural networks (MLPs) or bottleneck features [16, 17] generated by source language bottleneck networks are used as the input feature for a target language HMM/GMM acoustic model.
- *Automatic speech attribute transcription (ASAT)* [18–20]: speech attribute posteriors generated by the source language speech attribute detectors are used as the input features for a target language MLP.
- *Cross-lingual Kullback-Leibler based HMM (KL-HMM)* [21, 22]: each state in the KL-HMM is modeled by a discrete distribution to measure the KL-divergence between the state model and a source language phone posterior input.
- *Sequence-based phone mapping* [23, 24]: a source language phone sequence decoded by the source language phone recognizer is mapped to the target language phone sequence using a target language discrete HMM.
- *Posterior-based phone mapping* [25]: source language phone posteriors generated by the source language MLP are mapped to the target language phone posteriors using a target language MLP.

Although, the above methods offer some solutions to the under-resourced acoustic model development, this area of research remains active as building reliable acoustic models with limited training data has not been fully achieved. The objective of this thesis is to further investigate novel and efficient acoustic modeling techniques to improve performance of speech recognition for under-resourced languages.

1.1 Contributions

This thesis focuses on the third category of cross-lingual acoustic modeling which uses source acoustic models to generate cross-lingual features for the target language speech data. The reason is that it is a flexible approach which allows to use different architectures for source and target acoustic models. Three novel methods are proposed.

The first method is called the context-dependent phone mapping. This method is motivated by the fact that all human languages share similar acoustic space and therefore most sound units such as phones may be shared by different languages. Hence, a target language acoustic model can be mapped from acoustic models of other languages for the speech recognition purpose. Specifically, the source acoustic models are used as feature extractors to generate the source language phone posterior or likelihood scores. These scores are then mapped to the phones of the target language. Different to previous works which used context-independent monophone mapping, this work proposes context-dependent triphone states as the acoustic units to achieve higher mapping resolution. Experimental results in the Wall Street Journal (WSJ) corpus show that the proposed context-dependent phone mapping outperforms context-independent monophone mapping significantly, even in the case of very limited training data. In this study, two mapping functions, including simple linear mapping and nonlinear mapping with different training algorithms, are investigated. The results show that using the nonlinear mapping with the discriminative training criterion achieves the best performance.

While the first work focuses on improving the mapping resolution by using context information, the second work focuses on improving the cross-lingual features used for mapping, e.g. source language posterior feature using deep models. In this work, two approaches using deep neural networks (DNNs) are studied. In the first approach, DNNs are used as the acoustic model of the source language. It is hypothesized that DNNs can model the source language better than shallow models such as MLPs or GMMs, and the features generated by DNNs can produce better performance for phone mapping. In the second approach, DNNs are used to realize the phone mapping as opposed to the conventional MLP approach. Experimental results show that using DNN source acoustic models produce better results than shallow source acoustic models for cross-lingual phone mapping. Differently, using deep structures for phone mapping is only useful when a sufficient amount of target language training data is available.

The third contribution of the thesis is to apply exemplar-based model for acoustic modeling under limited training data conditions. Exemplar-based model is non-parametric and uses the training samples directly without estimating model parameters. This makes the approach attractive when training data are limited. In this study, a

specific exemplar-based model, called kernel density model, is used as the target language acoustic model. Specifically, the kernel density model uses cross-lingual bottleneck feature generated by the source language bottleneck DNN to estimate the likelihood probability of target language triphone states. In the kernel density model, the major design consideration is the choice of a distance function used to measure the similarity of a test sample and a training sample. In this work, a novel Mahalanobis-based distance metric learnt by minimizing the classification error rate on the training data is proposed. Experimental results show that the proposed distance produces significant improvements over the Euclidean distance metric. Another issue of the kernel density model is that it tries to estimate the distribution of the classes rather than optimal decision boundary between classes. Hence, its performance is not optimal in terms of speech recognition accuracy. To address this, a discriminative score tuning is introduced to improve the likelihood scores generated by the kernel density model. In the proposed score tuning, a mapping from the target language triphones to themselves is realized with the criterion of using the same principle of minimizing the training classification error.

The work on context-dependent phone mapping is published in the journal: IEICE Transactions on Information and Systems [26], and in the two conferences: IALP 2012 [27] and ISCSLP 2012 [28]. The work on applying deep neural networks on monolingual speech recognition and cross-lingual phone mapping is published in the two conferences: APSIPA ASC 2011 [29] and INTERSPEECH 2013 [30]. The work on the exemplar-based acoustic model is published the conference: INTERSPEECH 2014 [31].

1.2 Thesis Outline

This thesis is organized as follows:

In Chapter 2, the background information of statistical automatic speech recognition research is provided, followed by a review of the current state-of-the-art techniques for cross-lingual acoustic modeling.

In Chapter 3, a linear phone mapping is proposed for cross-lingual speech recognition where the target language phone model is formed as a linear combination of the source language phone models. To improve performance, a non-linear phone mapping realized by

a MLP is then applied. Finally, context-dependent phone mapping is proposed to achieve higher acoustic resolution as compared to the context-independent phone mapping for cross-lingual acoustic modeling.

In Chapter 4, deep neural networks (DNNs) are first investigated in monolingual speech recognition tasks. DNNs are then proposed to use for the cross-lingual phone mapping framework to improve the cross-lingual phone posteriors used for the mapping. In addition, DNNs are also investigated to realize the phone mapping function.

In Chapter 5, a non-parametric acoustic model, called exemplar-based model, is applied for speech recognition under limited training data conditions. To improve performance of the model, a novel Mahalanobis-based distance metric is then proposed. Finally, a score tuning is introduced at the top of the model to improve likelihood scores for decoding.

Finally, the thesis concludes in Chapter 6 with a summary of the contributions and several possible future research directions.

Chapter 2

Fundamentals and Previous Works

In this chapter, a brief introduction to typical speech recognition systems is provided, followed by a review of the current state-of-the-art techniques for cross-lingual acoustic modeling. The cross-lingual acoustic modeling techniques are grouped in three categories, i.e. universal phone set, subspace GMM, and using source language acoustic models as a feature extractor. The techniques in each category and sub-category will be reviewed individually.

As the review provided in the chapter is relatively extensive covering many techniques, to make it easier to appreciate the relationship between thesis's contributions to the existing techniques, it is necessary here to point out which techniques that are closely related to our study. The first contribution presented in Chapter 3 and second contribution presented Chapter 4 of the thesis are context-dependent phone mapping. The related techniques include posterior-based phone mapping (Section 2.2.3.4) and cross-lingual tandem (Section 2.2.3.1). The third contribution presented in Chapter 5 is exemplar-based acoustic modeling. The related literature is the exemplar-based method (Section 2.2.3.5).

2.1 Typical Speech Recognition System

Before examining different approaches in cross-lingual speech recognition, this section presents a brief introduction of the conventional ASR system. Fig. 2.1 shows the block diagram of a typical speech recognition system. It consists of five modules: feature extraction, acoustic model, language model, word lexicon, and decoding. These modules will be described in details in the next subsections.

Figure 2.1: Block diagram of a typical speech recognition system.

2.1.1 Feature extraction

The feature extraction module is used to generate speech feature vectors $\mathbf{X} = \{\mathbf{x}\}$ from the waveform speech signal s . The module's aim is to extract useful information and remove irrelevant information such as noise from the speech signal. The features are commonly computed on a frame-by-frame basis for every 10ms with a frame duration of 25ms. In such short frame duration, speech can be assumed to be stationary. Currently, the most popular features for ASR are Mel filterbank cepstral coefficients (MFCC) [32] and perceptual linear prediction (PLP) [33]. Normally, 39 dimensional MFCC or PLP feature vectors are used, consisting of 12 static features plus an energy feature, and the first and second time derivatives of the static features. The derivatives are included to capture the temporal information. A detailed discussion of features for ASR can be found in [34].

2.1.2 Acoustic model

The acoustic model is used to model the statistics of speech features for each speech unit such as a phone or a word. The Hidden Markov Model (HMM) [35] is the *de facto* standard used in the state-of-the-art acoustic models. It is a powerful statistical method to model the observed data in a discrete-time series.

An HMM is a structure formed by a group of states connected by transitions. Each transition is specified by its transition probability. The word *hidden* in HMMs is used to

indicate that the assumed state sequence generating the output symbols is unknown. In speech recognition, state transitions are usually constrained to be from left to right or self repetition, called the left-to-right model as shown in Fig. 2.2.

Figure 2.2: A left-to-right HMM model with three true states.

Each state of the HMM is usually represented by a Gaussian Mixture Model (GMM) to model the distribution of feature vectors for the given state. A GMM is a weighted sum of M component Gaussian densities and is described by Eq. (2.1).

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^M w_i g(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (2.1)$$

where $p(\mathbf{x}|\lambda)$ is the likelihood of a D -dimensional continuous-valued feature vector $\mathbf{x}$, given the model parameters $\lambda = \{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$, where w_i is the mixture weight which satisfies the constraint $\sum_i^M w_i = 1$, $\boldsymbol{\mu}_i \in \mathcal{R}^D$ is the mean vector, and $\boldsymbol{\Sigma}_i \in \mathcal{R}^{D \times D}$ is the covariance matrix of the i^{th} Gaussian function $g(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ defined by

$$g(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}_i|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \right). \quad (2.2)$$

The parameters of HMM/GMM acoustic models are usually estimated using the maximum likelihood (ML) criterion in speech recognition [35]. In ML, acoustic model parameters are estimated to maximize the likelihood of the training data given their correct word sequence. A comprehensive review of parameter estimation of the HMM/GMM model can be found in [35].

In the acoustic model of large-vocabulary continuous speech recognition (LVCSR) systems, an HMM is typically used to model a basic unit of speech called phone or phoneme. The mapping from words to phone sequences is captured in a word lexicon (pronunciation dictionary).

As pronunciation of a phone may vary significantly due to coarticulation effects, phones are expanded to include their contexts of the neighboring phones [36]. Typical acoustic models use triphones as the speech units by taking into account one neighboring phone in the left and right context. An example of triphones is illustrated in Fig. 2.3. In the triphone ‘i:-t+@’, /i:/ is the left context phone, /@/ is the right context phone, and /t/ is the central phone. The phones are represented following the International Phonetic Alphabet (IPA). The IPA represents the standard which has gained the widest acceptance among speech researchers for representing the range of sounds encountered across different languages¹.

Figure 2.3: Phonetic-acoustic modeling [1].

An issue to use context-dependent modeling such as triphones is data sparseness. The complexity of models increases exponentially with respect to the number of phones in context. For example, in the case of English, we have 40 monophones and hence there are a total of $40^3 = 64000$ possible triphones. However, many of them do not exist in the training data due to data sparseness or linguistic constraints. To address this problem,

¹International Phonetic Alphabet http://en.wikipedia.org/wiki/International_Phonetic_Alphabet

the decision tree-based clustering was investigated to tie triphones into physical models [36]. With this approach, the number of models is reduced and model parameters can be estimated robustly.

The latest state-of-the-art acoustic models have moved from HMM/GMM to HMM/DNN (Deep Neural Network) architecture [37–41]. DNNs were recently introduced as a powerful machine learning technique [42] and have been applied widely for acoustic modeling of speech recognition from small [37, 39] to very large tasks [40, 41] with a good success. In this thesis, application of DNNs for cross-lingual speech recognition is presented in Chapter 4.

2.1.3 Language model and word lexicon

While the acoustic model is used to score acoustic input feature, the language model assigns a probability to each hypothesized word sequence during decoding to include language information to improve ASR performance. For example, the word sequence “*we are*” should be assigned a higher probability than the sequence “*we is*” for an English language model. The typical language model is “ N -gram” [43]. In an N -gram language model, the probability of a word in a sentence is conditioned on the previous $N - 1$ words. If N is equal to 2 or 3, we have “bigram” or “trigram” language model, respectively. The paper [43] presents an excellent overview of recent statistical language models.

A word lexicon bridges the acoustic and language models. For instance, if an ASR system uses a phone acoustic model and a word language model, then the word lexicon defines the mapping between the words and the phone set. If a word acoustic model is used, the word lexicon is simply a one-to-one trivial mapping.

2.1.4 Decoding

The decoding block, which is also known as the search block, decodes the sequence of feature vectors into a symbolic representation. This block uses the acoustic model and the word lexicon to provide an acoustic score for each hypothesis word sequence W . The language model is simultaneously applied to compute the language model score for each hypothesized word sequence. The task of the decoder is to determine the best hypothesized word sequence that can then be selected based on the combined score

between the acoustic and language scores for the given input signal. This process is illustrated by the following equation:

$$\hat{W} = \arg \max_W p(W | \mathbf{X}) \quad (2.3)$$

where $\hat{W}$ is the word string that maximizes the posterior probability $p(W | \mathbf{X})$ given the input sequence of feature vectors $\mathbf{X} = \{\mathbf{x}\}$. Eq. (2.3) can be rewritten by applying the Bayes' formula as

$$\hat{W} = \arg \max_W \frac{p(\mathbf{X} | W)p(W)}{p(\mathbf{X})}. \quad (2.4)$$

As $p(\mathbf{X})$ is independent of the word sequence W , Eq. (2.4) can be reduced to

$$\hat{W} = \arg \max_W p(\mathbf{X} | W)p(W) \quad (2.5)$$

where $p(\mathbf{X} | W)$ is computed by the acoustic model, and $p(W)$ is computed by the language model.

To find the recognized word sequence efficiently, many search strategies have been proposed. The detailed description of various state-of-the-art search methods can be found in [44–46].

2.2 Acoustic Modeling for Under-resourced Languages

One major obstacle to build an acoustic model for a new language is that it is expensive to acquire a large amount of labeled speech data to train the acoustic model. To build a reasonable acoustic model for an LVCSR system, tens to hundreds of hours of training data are typically required. This constraint limits the application of traditional approaches especially for under-resourced languages.

The above challenge motivates speech researchers to investigate cross-lingual techniques that transfer acoustic knowledge from well-resourced languages to under-resourced languages [7]. Various cross-lingual acoustic modeling techniques have been proposed and can be grouped into the three categories:

- Universal phone set [7–10].
- Subspace Gaussian mixture model (SGMM) [4, 11, 12].

- Source language models act as a feature extractor [13–18, 20–25, 31].

The first category is referred to as universal phone set [7–10]. It generates a common phone set by pooling the phone sets of different languages (Fig. 2.4). A multilingual acoustic model can therefore be trained by using this common phone set. In general, with this approach, an initial acoustic model for a new language Y can be obtained by mapping from the multilingual acoustic model. Also, to improve performance on the target language, the initial acoustic model is refined using adaptation data of the target language.

Figure 2.4: An example of the universal phone set approach.

The second category is the cross-lingual subspace Gaussian mixture model (SGMM) [4, 11, 12]. As shown in Fig. 2.5, in the SGMM acoustic model, the model parameters are separated into two classes, i.e. language-independent parameters and language-dependent parameters. With such a model, the language-independent parameters from a well-resourced (source) language can be reused and the language-dependent parameters are trained with speech data from the target language. As language-dependent parameters only account for a small proportion of the overall parameters, they could be reliably trained with a small amount of training data. Hence the SGMM is a possible approach to supporting speech recognition development of under-resourced languages.

In the third category, which is also the most popular approach, the source acoustic model is used as a feature extractor to generate cross-lingual features for target language

Figure 2.5: Illustration of the cross-lingual subspace Gaussian mixture model (SGMM) approach.

speech data (Fig. 2.6). As these cross-lingual features are higher-level and more meaningful features as compared to conventional features such as MFCCs, they allow the use of a simpler model that could be trained using only a small amount of training data for target acoustic model development.

Figure 2.6: Source acoustic models act as a feature extractor.

In the next subsections, the above cross-lingual techniques will be reviewed in details.

2.2.1 Universal phone set method

In this approach, a common phone set is generated by pooling the phone sets of different languages to train a multilingual acoustic model [7, 8]. This allows the model to be shared by various languages and hence reduces the complexity and number of parameters for the multilingual LVCSR system. The acoustic model for a new language can be achieved by mapping from the multilingual acoustic model. Also, to improve performance, this initial model is then bootstrapped using the new language training data.

The rest of the section is organized as follows. Subsection 2.2.1.1 presents the one-to-one phone mapping techniques to map the phone sets of different languages. Subsection 2.2.1.2 discusses about building a multilingual acoustic model. Subsection 2.2.1.3 presents recent techniques for acoustic model refinement.

2.2.1.1 One-to-one phone mapping

One-to-one phone mapping is used to map a phone in a language to a phone of another language. Phone mapping is used to pool the phone sets of different languages when building a multilingual acoustic model. In addition, phone mapping is also used to map the multilingual acoustic model to the target language acoustic model. Phone mapping can be grouped into two categories: the knowledge-based approach and the data-driven approach.

In the knowledge-based approach [2], a human expert defines the mapping between the phone sets of two languages by selecting the closest IPA counterpart. Hence, no training data is required in the knowledge-based phone mapping. Fig. 2.7 shows the phone sets of Spanish and Japanese in the IPA format.

Phone Category		Spanish	Japanese
Stops	Voiced	/b/,/g/,/d/	/b/,/g/,/d/
	Unvoiced	/p/,/k/, /t/	/p/,/k/, /t/
Fricatives	Voiced	/ð/, /ʒ/, /z/,/β/	/z/
	Unvoiced	/f/, /h/, /s/	/f/,/h/, /s/, /ʃ/
Affricatives	Voiced	-	/dʒ/
	Unvoiced	/tʃ/	/ts/, /tʃ/
Nasals		/ɲ/, /n/, /m/	/ɲ/,/n/,/n̠/,/m/
Diphthongs		-	/ie/
Glides		/j/, /w/	/j/, /w/
Liquids		/r/, / ɾ /, /l/	/r/
Vowels		/a/, /e/, /i/, /o/, /u/	/a/, /e/, /e:/, /i/,/i:/, /o/,/o:/, /u/, /u:/
Silence, Noise Glottals		sp, sil	sp, sil, /ʔ/
Total Phones		29	34

Figure 2.7: Phones of Spanish and Japanese are expressed in the IPA format [2].

In data-driven mapping approach, a mathematically tractable, predetermined performance measure is used to determine the similarity between acoustic models from different languages [2]. The two popular performance measures are used: confusion-based matrices and distance-based measures [47, 48].

The confusion matrix technique involves the use of a grammar-free phone recognizer of the source language to decode the target speech. The hypothesized transcription is then cross compared with the target language reference transcription. Based on the frequency of co-occurrence between the phones in the source and target inventories, a confusion matrix is generated. The one-to-one phone mapping to a target phone is derived by picking the hypothesized phone which has the highest normalized confusion score.

An alternative data-driven method is to estimate the difference between phone models using the model parameters directly. The distance metrics between phone models which have been used include Mahalanobis, KullBack-Liebler and Bhattacharya [49].

Experiments on the Global Phone corpus in [2] showed that in most cases, the two data-driven mapping techniques produce similar mappings. Comparing to the knowledge-based phone mapping, the two data-driven techniques provide a modest improvement when mapping is conducted for the matched conditions. However, the result showed that knowledge-based mapping is superior when mappings are required in the mismatched conditions.

2.2.1.2 Multilingual acoustic models

In [8], the multilingual acoustic model is built based on the assumption that the articulatory representations of phones are similar across languages and phones are considered as language-independent units. This allows the model to be shared by various languages and hence reduces the complexity and number of parameters for the multilingual LVCSR system. Two combination methods ML-mix and ML-tag are proposed to combine the acoustic models of different languages [8].

Denote $p(\mathbf{x}|s_i)$ to be the emission probability for feature vector $\mathbf{x}$ of state s_i ,

$$p(\mathbf{x}|s_i) = \sum_{k=1}^{K_i} c_{s_i,k} N(\mathbf{x}|\boldsymbol{\mu}_{s_i,k}, \boldsymbol{\Sigma}_{s_i,k}) \quad (2.6)$$

where s_i is a state of a phone in a specific language, $N(\mathbf{x}|\boldsymbol{\mu}_{s_i,k}, \boldsymbol{\Sigma}_{s_i,k})$ is the normal distribution with mean vector $\boldsymbol{\mu}_{s_i,k}$ and covariance matrix $\boldsymbol{\Sigma}_{s_i,k}$. $c_{s_i,k}$ is the mixture weight for mixture k , K_i is the number of mixtures for state s_i .

In the ML-mix combination, the training data are shared across different languages to estimate the acoustic model's parameters. The phones in different languages which have the same IPA unit are merged and the training data of languages belonging to the same IPA unit are used to train this universal phone, i.e. during the training process, no language information is preserved. The ML-mix model combination method can be described by:

$$\text{ML - mix : } \begin{cases} c_{s_i,k} = c_{s_j,k}, \forall i, j : IPA(s_i) = IPA(s_j), \\ \boldsymbol{\mu}_{s_i,k} = \boldsymbol{\mu}_{s_j,k}, \forall i, j : IPA(s_i) = IPA(s_j), \\ \boldsymbol{\Sigma}_{s_i,k} = \boldsymbol{\Sigma}_{s_j,k}, \forall i, j : IPA(s_i) = IPA(s_j), \end{cases}$$

Different to the above method, the ML-tag combination method preserves each phone's language tag. Similar to ML-mix, all the training data and the same clustering procedure are used. However, only the Gaussian component parameters are shared across languages, the mixture weights are different. This approach can be described by:

$$\text{ML - tag : } \begin{cases} c_{s_i,k} \neq c_{s_j,k}, \forall i \neq j, \\ \boldsymbol{\mu}_{s_i,k} = \boldsymbol{\mu}_{s_j,k}, \forall i, j : IPA(s_i) = IPA(s_j), \\ \boldsymbol{\Sigma}_{s_i,k} = \boldsymbol{\Sigma}_{s_j,k}, \forall i, j : IPA(s_i) = IPA(s_j), \end{cases}$$

Experiments were conducted in [9] to evaluate the usefulness of two above approaches. Five languages were used, Croatian, Japanese, Korean, Spanish, and Turkish. In this work, the authors concentrated on recognizing these five languages simultaneously which are involved for training the multilingual acoustic models. The ML-tag combination method outperforms ML-mix in all languages. In addition, the ML-tag model reduces 40% number of parameters as compared to the monolingual systems. Although the ML-tag model performs slightly worse than the language dependent acoustic models, such model can be used to rapidly build an acoustic model for a new language [9, 10] using different model refinement techniques which will be discussed in the next section.

2.2.1.3 Cross-lingual model refinement

To build a new language acoustic model from the pre-trained multilingual acoustic model, a phone mapping from the given multilingual model phone set to the new language phone

set is first identified. After that the initial acoustic model of the target language is generated by copying the required phone models from the multilingual acoustic model. The performance of such initial model is usually very poor. To improve, the initial target model is adapted using target language speech data. The rest of this section discusses two popular methods for cross-lingual model refinement: model bootstrapping and model adaptation.

Model bootstrapping is the simplest cross-lingual model refinement approach [8, 50, 51]. Bootstrapping simply means that the model is completely re-trained with the target language training data using the initial acoustic model. Although experimental results have shown that there is no significant improvement over monolingual training of the target language, there is a significant improvement in convergence speed of the training process [8, 50, 51]

When the amount of target training data is scarce, the traditional model adaptation techniques can be applied to the initial acoustic model. Model adaptation techniques can be grouped into two main categories: direct and indirect adaptation. In direct adaptation, the model parameters are re-estimated given the adaptation data. Bayesian learning, in the form of maximum a posteriori (MAP) adaptation [52, 53], forms the mathematical framework for this task. In contrast, the indirect adaptation approach avoids the direct re-estimation of the model parameters. Mathematical transformations are built to convert the parameters of the initial model to the target conditions. The most popular representative of this group is maximum likelihood linear regression (MLLR) [54] which transforms the input features to maximize the likelihood of the acoustic model.

The first study using model adaptation scheme for cross-lingual model refinement was reported in [55]. In that study, online MAP adaptation was used to adapt monolingual and multilingual German, US-English, US-Spanish acoustic models to Slovenian. Only the means of the Gaussians were adapted resulting in an absolute gain in WER up to 10% for a digit recognition task.

Methods using offline MAP adaptation were reported in [56, 57]. Multilingual English-Italian-French-Portuguese-Spanish models were transformed to German in [56]. An improvements up to 5% in WER over mapped, non-adapted models was achieved on a isolated word recognition task. In [57], the language transfer was applied from English to

Mandarin. In these experiments the phone recognition could be improved by 10% using MAP adaption.

In [47], Czech was used as the target language while Mandarin, Spanish, Russian, and English were the source languages. Adaptation to Czech was performed by concatenating MLLR and MAP. Although significant improvements were achieved over the non-adapted models, the performance of a pure Czech triphone system trained with the adaptation data was not reached.

To sum up, the universal phone mapping is an intuitive approach for cross-lingual acoustic modeling which is based on the assumption that phones can be shared in different languages. In this approach, a multilingual acoustic model is built by pooling the phone sets of different languages to train a multilingual acoustic model. The acoustic model for a new language is achieved by using a one-to-one phone mapping from the multilingual acoustic model. However, phones in different languages can be similar, they are unlikely to be identical. Hence, the one-to-one phone mapping may cause the poor performance of the initial cross-lingual acoustic model. In Chapter 3, we propose a novel many-to-one phone mapping to improve speech recognition of under-resourced languages.

2.2.2 Cross-lingual subspace GMM (SGMM)

Unlike the universal phone set approach, in the subspace GMM (SGMM) approach, a novel model structure is proposed to reduce the amount of required training data. In this approach [4, 12], the model parameters are separated into two classes, i.e. the subspace parameters that are almost language independent, and phone state specific parameters that are language dependent. As well-resourced languages can first be used to train the subspace parameters and the limited target language data used to train phone state parameters. As phone state parameters only account for a small proportion of the overall model, they could be reliably trained using a limited amount of target training data.

Similar to the conventional GMM model, the SGMM also uses mixtures of Gaussians as the underlying state distribution, however, parameters in the SGMM are shared between states. Sharing is justified by the high correlation between states distributions since the variety of sounds that the human articulatory tract can produce is limited [58].

In the SGMM model [58], the distribution of the features in HMM state j is a mixture of Gaussians:

$$p(\mathbf{x}|j) = \sum_{i=1}^K w_{ji} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{ji}, \boldsymbol{\Sigma}_i) \quad (2.7)$$

where, there are K full-covariance Gaussians shared between all states. Unlike the conventional GMM, in the SGMM, the state dependent mean vector $\boldsymbol{\mu}_{ji}$ and mixture weight w_{ji} are not directly estimated as parameters of the model. Instead, $\boldsymbol{\mu}_{ji}$ of state j is a projection into the i^{th} subspace defined by a linear subspace projection matrix $\mathbf{M}_i$,

$$\boldsymbol{\mu}_{ji} = \mathbf{M}_i \mathbf{v}_j \quad (2.8)$$

where $\mathbf{v}_j$ is the state projection vector for state j and a language-dependent parameter. The subspace projection matrix $\mathbf{M}_i$ is shared across state distribution and language-independent which has the dimension of $D \times S$ where D is the dimension of input feature vector $\mathbf{x}$, S is the dimension of the state projection vector $\mathbf{v}_j$. The mixture weight w_{ji} in Eq. (2.7) is derived from the state projection vector $\mathbf{v}_j$ using a log-linear model,

$$w_{ji} = \frac{\exp(\mathbf{q}_i^T \mathbf{v}_j)}{\sum_{i'=1}^K \exp(\mathbf{q}_{i'}^T \mathbf{v}_j)} \quad (2.9)$$

with a globally shared parameter $\mathbf{q}_i$ determining the mapping.

To have more flexibility for the SGMM model, the concept of substates is adopted for state modeling. Specifically, the distribution of a state j can be represented by more than one vector $\mathbf{v}_{jm}$, in this case, m is the substate index. Similar to the state distribution in Eq. (2.7), substate distribution is again a mixture of Gaussians. The state distribution is then a mixture of substate distributions,

$$p(\mathbf{x}|j) = \sum_{m=1}^{M_j} c_{jm} \sum_{i=1}^K w_{jmi} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{jmi}, \boldsymbol{\Sigma}_i) \quad (2.10)$$

$$\boldsymbol{\mu}_{jmi} = \mathbf{M}_i \mathbf{v}_{jm} \quad (2.11)$$

$$w_{jmi} = \frac{\exp(\mathbf{q}_i^T \mathbf{v}_{jm})}{\sum_{i'=1}^K \exp(\mathbf{q}_{i'}^T \mathbf{v}_{jm})} \quad (2.12)$$

Table 2.1: Performance and model complexity of GMM and SGMM models in a monolingual task [4].

Model	Number of Sub-states	Number of parameters		WER(%)
		State-independent	State-specific	
GMM	n.a	0	2427k	52.5
SGMM	1.9k	952k	77k	48.9
SGMM	12k	952k	492k	47.5

where c_{jm} is the relative weight of substate m in state j .

Table 2.1 shows the number of parameters and word error rate (WER) for GMM and SGMM models in a monolingual task [4]. In the conventional GMM model, all parameters are state-dependent. In contrast, a significant amount of parameters are shared between states in the SGMM models. The number of state-dependent parameters in SGMMs varies following the number of sub-states. Although, in this case, the SGMM models have a smaller number of total parameters as compared to the GMM model, the WERs in both the SGMM models are significantly better.

With its special architecture, the SGMM can be very naturally applied for cross-lingual speech recognition. The state independent SGMM parameters, Σ_i , $\mathbf{M}_i$, $\mathbf{q}_i$ can be trained from well-resourced languages while the state dependent parameters $\mathbf{v}_{jm}$, c_{jm} are trained from limited training data of the under-resourced language.

Study in [4] is the first work applying SGMM for cross-lingual speech recognition. In that work, the SGMM shared parameters are trained from 31 hours of Spanish and German while the state-specific parameters are trained on 1 hour of English which is randomly extracted from the Callhome corpus. With this approach, the cross-lingual SGMM model achieves 11% absolute improvement over the conventional monolingual GMM system. The idea of SGMM is further developed using the regularized SGMM estimation approach [59]. The experimental result on the GlobalPhone corpus showed that regularizing cross-lingual SGMM systems (using the ℓ_1 -norm) results in a reduced WER.

One issue when using multiple languages to estimate the shared parameters in the SGMM is the potential mismatch of the source languages with the target language, e.g. the differences in phonetic characteristics, corpus recording conditions, and speaking styles. To address this, maximum a posteriori (MAP) adaptation can be used [12]. In

particular, the target language model can be trained by MAP adaptation on the phonetic subspace parameters. This solution results in a consistent reduction in word error rate on the GlobalPhone corpus [12].

In conclusion, in the SGMM approach, a novel acoustic model structure is proposed to reduce the amount of training data required for acoustic model training. In the SGMM, model parameters are separated into two classes: language-dependent and language-independent. With such a model, language-independent parameters can be borrowed from the well-resourced language and the limited target language data are used to train the language-dependent parameters. Although, the SGMM approach can provide significant improvements over the monolingual models, this approach requires the source and target models to have the similar structures. This limits the flexibility of the approach for different cross-lingual tasks. In the next section, a more flexible cross-lingual framework will be discussed.

2.2.3 Source language models act as a feature extractor

This section discusses about the third category which is also currently the most popular approach. In this approach, the idea is to use source language acoustic models as a feature extractor to generate high-level, meaningful features to support under-resourced acoustic model development. As listed in Table 2.2, various cross-lingual methods have been proposed using different types of target acoustic models and different types of cross-lingual features. These methods will be presented in details in the following subsections.

The proposed works in this thesis belong to this category. The works in Chapter 3 and Chapter 4 are cross-lingual phone mapping. The related techniques include posterior-based phone mapping (Subsection 2.2.3.4) and cross-lingual tandem (Subsection 2.2.3.1). The works in Chapter 5 are exemplar-based acoustic modeling. The related literature is the exemplar-based method (Subsection 2.2.3.5).

2.2.3.1 Cross-lingual tandem approach

The tandem approach was proposed by Hermansky *et al.* in 2000 for monolingual speech recognition. In this approach, a neural network is used to generate phone posterior feature for an HMM/GMM acoustic model. This feature takes advantages of the discriminative

Table 2.2: Different cross-lingual methods using the source acoustic models as a feature extractor.

Method	Cross-lingual feature	Target acoustic model
Cross-lingual tandem	Phone posteriors [13–15] Bottleneck feature [16, 17]	HMM/GMM
ASAT [18–20]	Speech attribute posteriors	HMM/MLP
Cross-lingual KL-HMM [21, 22]	Phone posteriors	KL-HMM
Sequence-based phone mapping [23, 24]	Phone sequences	Discrete HMM
Posterior-based phone mapping [25]	Phone posteriors	HMM/MLP
Cross-lingual exemplar-based model [31]	Bottleneck features	Exemplar-based model

ability of neural networks while retaining the benefit of conventional HMM/GMM models such as model adaptation.

The tandem approach was recently applied for cross-lingual tasks [13–17]. As shown in Fig. 2.8, in the cross-lingual tandem approach, the source language data are used to train an MLP. This MLP is then used to generate source language phone posteriors for the target language data. These posterior scores are passed to a *log* function. Taking *log* renders the feature distribution to appear more Gaussian [60]. As the posterior features are of high dimension, they are not suitable for the HMM/GMM modeling. To reduce the dimensionality of the feature, a dimensionality reduction such as PCA (Principal Component Analysis) or HLDA (Heteroscedastic Linear Discriminant Analysis) is applied. To further improve performance, these features are concatenated with low level spectral features such as MFCCs or PLPs to create the final feature vector which is modeled by the target language’s HMM/GMM.

The main disadvantage of the tandem approach comes from its dimensionality reduction process as it results in information loss. This effect will be investigated in Chapter 3, and Chapter 4 of this thesis.

A variation of the tandem approach is the bottleneck feature approach [61]. The bottleneck feature is generated using an MLP with several hidden layers where the size of the middle hidden layer, i.e. bottleneck layer is set to be small. With this structure, an arbitrary feature size can be obtained without using dimensionality reduction step and is independent to the MLP training targets. The bottleneck feature has been used widely in speech recognition and provides a consistent improvement over conventional

Figure 2.8: Cross-lingual tandem approach.

features such as MFCCs, PLPs and posterior features as well. In [16, 17], bottleneck feature has been applied for cross-lingual ASR by using a bottleneck MLP trained for source languages to generate the bottleneck features for a target language HMM/GMM model. Their results showed a potential ability to use bottleneck feature in cross-lingual speech recognition. In Chapter 4 and Chapter 5 of this thesis, cross-lingual bottleneck feature will be used in the proposed cross-lingual framework.

2.2.3.2 Automatic speech attribute transcription (ASAT)

Automatic speech attribute transcription (ASAT) proposed by Lee *et al* [19] is a novel framework for acoustic modeling that uses a different type of speech units called speech attributes. Speech attributes (SAs) such as voicing, nasal, dental, describe how speech is articulated. They are also called phonological features, articulatory features, or linguistic features. SAs have been shown to be more language universal than phones and hence may be used as speech units and be adapted to a new language more easily [62, 63].

The set of SAs can be divided into subsets as illustrated in Table 2.3 [5]. SAs in each subset can then be classified by an MLP to estimate posterior probability $p(a_i|\mathbf{x})$, where a_i denotes the i^{th} SA in that subset, and $\mathbf{x}$ is the input vector at time t . Experimental results with monolingual speech recognition in [5] showed that SAs help to improve performance of a continuous numbers recognition task under clean, reverberant and noisy conditions.

In the ASAT framework [19], SA posteriors generated by various detectors are merged to estimate higher level speech units such as phones or words to improve decoding. The framework exploits language-independent features by directly integrating linguistic

Table 2.3: Speech attribute subsets [5].

Attribute subset	Possible output value
Voicing	+voice, -voice, silence
Manner	stop, vowel, fricative, approximant, nasal, lateral, silence
Place	dental, labial, coronal, palatal, velar, glottal, high, mid, low, silence
Front-Back	front, back, nil, silence
Lip Rounding	+round, -round, nil, silence

knowledge carried by SAs into the ASR design process. This approach allows to build a speech recognition system for a new target language with a small amount of training data.

Fig. 2.9 shows the ASAT framework for cross-lingual speech recognition. SA detectors are used to generate SA posterior probability $p(a_i|\mathbf{x})$. An event merger is then used to combine these posteriors to generate target phone posterior probability $p(p_j|\mathbf{x})$. Finally, these phone posteriors are used by a decoder (evidence verifier) as in the hybrid ASR approach [64].

Figure 2.9: Automatic speech attribute transcription (ASAT) framework for cross-lingual speech recognition.

In our recent work [65], the experiments also showed the usefulness of SAs in a cross-lingual ASR framework. Specifically, SA posterior probabilities estimated by the SA

detectors are mapped to context-dependent tied states of the target language by an MLP. The results showed that using SAs alone provides lower performance over the phone-based approach, however, when SA posteriors are concatenated with phone likelihood scores generated by the source HMM/GMM model, consistent improvements are observed over both the individual systems. This shows that SAs provide complementary information to phone-based scores for cross-lingual modeling.

In [66], we proposed a method to recognize SAs more accurately by considering the left or the right context of SAs called bi-SAs. Our results on the TIMIT database showed that the higher resolution SA can help to improve SA and phone recognition performance significantly. The idea of context-dependent bi-SAs was then applied on cross-lingual ASR tasks [65]. Specifically, four bi-SA detectors: left-bi-manner, right-bi-manner, left-bi-place and right-bi-place are trained with source language data. Given a target language speech frame, each detector then generates posterior probabilities for each class of bi-SAs. These posteriors are further combined and mapped into target triphone states. Experimental results showed that context-dependent SAs perform better over context-independent SAs in cross-lingual tasks.

2.2.3.3 Kullback-Leibler based HMM (KL-HMM)

In this subsection, the KL-HMM, a special type of HMM based on the Kullback-Leibler divergence is presented [21, 22, 67, 68]. Similar to the tandem approach, the KL-HMM uses phone posteriors generated by the source language MLP as the input feature. However, while the tandem approach uses a mixture of Gaussians to model the statistics of the feature, the KL-HMM has a simpler architecture with less parameters and hence can be estimated more robustly in the case of limited training data.

In the KL-HMM, state emission probability $p(\mathbf{z}_t | s_d)$ of state s_d for feature $\mathbf{z}_t \in \mathcal{R}^K$ is simply modeled by a categorical distribution² $\mathbf{y}_d = [y_d^1, \dots, y_d^K]^T$, $d \in \{1, \dots, D\}$, where K is the dimensionality of the input feature and D is the number of states in KL-HMM. The input feature vector $\mathbf{z}_t = [z_t^1, \dots, z_t^K]^T$ is the phone posterior probability generated by an MLP and satisfies the constraint:

²a categorical distribution is a probability distribution that describes the result of a random event that can take on one of K possible outcomes, with the probability of each outcome separately specified.

$$\sum_{i=1}^K z_t^i = 1. \quad (2.13)$$

The posterior vector $\mathbf{z}_t$ is modeled as a discrete probability distribution on the HMM state space. For each state s_d and each feature vector $\mathbf{z}_t$, a dissimilarity measurement between the two discrete distributions $\mathbf{y}_d$ and $\mathbf{z}_t$ can be measured by their KL-divergence

$$KL(\mathbf{y}_d || \mathbf{z}_t) = \sum_{k=1}^K y_d^k \log \frac{y_d^k}{z_t^k}. \quad (2.14)$$

In the conventional HMM/GMM model, if we ignore the effect of state transition probabilities and language model factor, the cost function during decoding to find the optimum word sequence W can be expressed by

$$J_W(\mathbf{X}) = \min_{\Phi^W} \sum_{t=1}^T -\log p(s_{\Phi_t} | \mathbf{x}_t) \quad (2.15)$$

where Φ^W represents all possible state sequences of length T allowed by W , $\mathbf{X} = \{\mathbf{x}_t\}$ is the sequence of the input feature vectors. If we use the KL divergence approach as in Eq. (2.14), we can rewrite Eq. (2.15) as

$$J_W(\mathbf{X}) = \min_{\Phi^W} \sum_{t=1}^T KL(\mathbf{y}_{\Phi_t} || \mathbf{z}_t). \quad (2.16)$$

The model described by Eq. (2.16) can be interpreted as a finite state machine where each state has an associated cost given by the KL-divergence between its corresponding target distribution, $\mathbf{y}_d$ and the posterior vector, $\mathbf{z}_t$ [68]. Hence, the parameters of the KL-HMM are simply the D state distributions $\{\mathbf{y}_d\}_{d=1}^D$. The number of parameters in this model is very small as compared to the HMM/GMM model and hence the KL-HMM is a good option for acoustic modeling under limited training data conditions.

In the training process, the arithmetic mean of all the posterior features associated with KL-HMM state s_d forms state categorical distribution $\mathbf{y}_d$.

$$\mathbf{y}_d = \frac{1}{N_d} \sum_{i=1}^{N_d} \mathbf{z}_i \quad (2.17)$$

where N_d is the number of training samples $\mathbf{z}_i$ which are associated with state s_d .

Experiments for the monolingual tasks conducted in [67] indicated that the KL-HMM outperforms the tandem method for a small task with limited training data. However, one weak point of the KL-HMM approach is that the model is too simple, i.e. it has very few parameters and hence could not achieve better performance when more training data become available. This observation was seen in [68] where Aradilla *et al.* applied the KL-HMM to large vocabulary for the WSJ corpus. Experimental results showed that KL-HMM performs slightly worse than the tandem approach.

In [21, 22], Imseng *et al.* applied the KL-HMM approach for cross-lingual tasks. Similar to the cross-lingual tandem [13–15] and posterior-based phone mapping [25] approaches, an MLP well-trained by a source language is used to generate phone posterior probabilities for speech data of the target language. These posteriors are then used as the input feature for the target language acoustic model such as HMM/GMM (tandem), HMM/MLP (phone mapping) or KL-HMM. Experiments in [21] indicated that the cross-lingual KL-HMM achieves better performance than the cross-lingual tandem model if less than 4 hours of target data are used. However, when more training data became available, the cross-lingual KL-HMM does not continue to improve, and is even worse than the monolingual HMM/GMM model. This phenomenon is also repeated in [22], when more than 1 hour of target language training data are used, the KL-HMM performs worse than other models such as cross-lingual SGMM or HMM/GMM.

In conclusion, the KL-HMM approach has advantages such as its simple architecture with a small number of parameters. Consequently, it also means that it is useful only for limited training data conditions. The followings summarize the disadvantages of the KL-HMM approach.

- Lack of flexibility: the number of parameters in KL-HMM depends only on the number of outputs of the source language MLP and number of states in the target language. Normally, when the amount of target training data changes, the complexity of the target model should be changed accordingly. However, as the number of states in the target language is constraint by the limited target training data while a change in the number of the MLP outputs results in re-training the source MLP.

- The KL-HMM modeling does not allow easy implementation for additional features such as non-probabilistic features. In addition, as the input feature of KL-HMM depends directly on the MLP, if the MLP is not well-trained, the KL-HMM will not work well.

2.2.3.4 Sequence-based and posterior-based phone mapping

In the phone mapping approach, the speech data of the target language are first recognized as phone sequences [23] or phone posteriors [25] of a source language. These phone sequences or posteriors are then mapped to the corresponding phone sequences or posteriors of the target language followed by the standard decoding process.

The cross-lingual phone recognition process can be formulated as follows [23, 24, 69]:

$$\hat{Y} = \arg \max_Y \{p(Y|\mathcal{O}, \mathcal{X}, \mathcal{Y}, \theta(\mathcal{X}))\} = \arg \max_Y \left\{ \sum_X p_{\mathcal{M}}(Y|X) p(X|\mathcal{O}, \mathcal{X}, \theta(\mathcal{X})) \right\} \quad (2.18)$$

where $\mathcal{O}$ is the observation sequence, $\mathcal{X}$ and $\mathcal{Y}$ are the source and target phone sets, respectively, $\theta(\mathcal{X})$ denotes an acoustic model for $\mathcal{X}$ and $p_{\mathcal{M}}(Y|X)$ is the probability of mapping from phone sequence X to phone sequence Y . This can be approximated as a two-stage process:

$$\hat{X} = \arg \max_X \{p(X|\mathcal{O}, \mathcal{X}, \theta(\mathcal{X}))\} \quad (2.19)$$

$$\hat{Y} \approx \arg \max_Y \left\{ p_{\mathcal{M}}(Y|\hat{X}) \right\} \quad (2.20)$$

The first stage is simply the decoding process using the source phone recognizer, $\theta(\mathcal{X})$ to obtain the best source phone sequence $\hat{X}$ given $\mathcal{O}$. Then, in the second stage, the mapping between the source language phone sequence, $\hat{X}$ to the target language phone sequence, $\hat{Y}$ is implement by a mapping $\mathcal{M}$: $\mathcal{X} \rightarrow \mathcal{Y}$.

In [23], Sim and Li proposed a novel sequence-based phone mapping method where the mapping function $\mathcal{M}$ is modeled as a probabilistic model. The mapping model is estimated as follows:

$$\hat{\mathcal{M}} = \arg \max_{\mathcal{M}} \left\{ \sum_{Y \in \Psi} p(Y|X, \mathcal{O}, \mathcal{M}) \right\} \quad (2.21)$$

where Ψ denotes the target language training set, X and Y are the pair of phone sequences for the speech utterance, $\mathcal{O}$. In [23, 24] the mapping between two phone sequences is implemented as a discrete HMM where each state in the HMM model represents a phone in the target language while the phone sequence generated by the source phone recognizer X is used as the input for the discrete HMM. The HMM can be trained using a maximum likelihood criterion. However, in the generative HMM model, it is required $|X| \geq |Y|$ where $|X|, |Y|$ are the length of the two phone sequences X, Y , respectively. To satisfy this condition, in [23, 24], Sim and Li expanded the original phone sequence X into smaller “frames” based on the duration of each phone in X . This process is illustrated in Fig. 2.10.

Figure 2.10: Probabilistic sequence-based phone mapping model.

The experiments conducted in the SpeechDat-E and TIMIT databases [23, 24] indicated that the probabilistic sequence-based phone mapping model can be trained with a small amount of training data due to the small number of parameters in the discrete HMM.

A main limitation of the sequence-based phone mapping system is that only the 1-best phone sequence $\hat{X}$ given by the source phone recognizer is used Eq. (2.19, 2.20) as the input for mapping. That means that other possibilities in the decoding are ignored. This can be interpreted as information loss by “quantization effect” during the decoding

process. To address this, in [25], the author suggested that the mapping between the source and target phone sets can be implemented before the decoding stage, i.e. at the probability level. Specifically, the source phone posteriors generated by the source MLP acoustic model are used as the input for the mapping to the target phone posteriors. The mapping is implemented using a product-of-expert method where either the experts or their weights are modeled as functions of the posterior probabilities of the source phones, generated by the source phone MLP. Experimental results on the NTIMIT database showed that using only 15.6 minutes of training data to train the posterior-based phone mapping achieves a cross-lingual phone recognition errors of 46.0% while the monolingual phone recognizer trained on the same amount of data obtained 53.4% phone error rate.

Based on the spirit of this approach, the thesis proposes an extension called “context-dependent phone mapping” and applies it for large vocabulary word recognition tasks. The result will be presented in Chapter 3.

2.2.3.5 Exemplar-based approach

While the above methods use parametric models such as GMM, MLP, KL-HMM, or discrete HMM for the target acoustic modeling, this section introduces the use of non-parametric models called exemplar-based model that can use the training samples directly. Unlike parametric methods, exemplar-based methods, such as k-nearest neighbors (k-NN) [70] for classification and kernel density (or Parzen window) [70] for density estimation, do not assume a parametric form for the discriminant or density functions. This makes them attractive when the functional form or the distribution of decision boundary is unknown or difficult to learn and especially when training data are limited.

In the machine learning community, two broad categories of approaches for modeling the observed data are suggested [70]. The first category is referred to as eager (offline) learning that uses all the available training data to build a model before the test sample is seen. The second category is referred to as lazy (exemplar-based) learning that selects a subset of exemplars from the training data to build a local model for each test sample. Although exemplar-based methods have been popular in many tasks such as face recognition [71], object recognition [72], audio classification [73], they are only recently applied for speech recognition [74–76].

An exemplar-based speech recognition system typically contains three stages: exemplar selection, instance modeling, and decoding [77].

In the exemplar selection stage, the process identifies the most relevant training instances to each test instance using a distance metric. Given a suitable distance metric, different types of exemplars can be used e.g. individual frames [74], or segmental level such as fixed-length sequences of multiple frames [78] or variable length segments [79].

In the instance modeling stage, the training exemplars identified via the first stage as most relevant for the test instance are used to model the test instance. For instance modeling, each training exemplar is associated with a class label that can range from HMM-states on individual frame [74], to word labels for word segments [79]. For example, in the k-nearest neighbor (k-NN) method, the class posterior of class q_i given test instance $\mathbf{x}_t$ can be estimated as

$$p(q_i|\mathbf{x}_t) = \frac{N_{it}}{N_t} \quad (2.22)$$

where N_t is the total number of the most relevant training exemplars for test instance $\mathbf{x}_t$, N_{it} is the number of the most relevant training exemplars belong to class q_i .

The third stage is to perform decoding to recognize the test utterance. Specifically, given the observation $X = \{\mathbf{x}_t\}$, the instance models from the previous stage, and a set of subword units U , an acoustic score $p(X|U)$ is computed directly for the segmental approaches. For the frame-based techniques, class posterior $p(q_i|\mathbf{x}_t)$ needs to be converted to the frame-based observation likelihood to compute acoustic score $p(X|U)$ of the whole utterance.

Recently, several studies have successfully applied the exemplar-based approach for acoustic modeling [74–76]. In [74], the authors proposed to use a nearest neighbor classifier at the frame level in a speech recognition system. Specifically, the kernel density estimation [80] is used to replace GMM in the HMM/GMM model for state likelihood estimation as:

$$p(\mathbf{x}_t|q_i) = \frac{1}{N_{it}} \sum_{j=1}^{N_{it}} \exp\left(-\frac{\|\mathbf{x}_t - \mathbf{e}_{ij}\|^2}{\sigma}\right) \quad (2.23)$$

where $\mathbf{x}_t$ is the test feature vector at frame t , $\mathbf{e}_{ij}$ is the j^{th} exemplar of class q_i , $\|\mathbf{x}_t - \mathbf{e}_{ij}\|^2$ denotes the distance between $\mathbf{x}_t$ and $\mathbf{e}_{ij}$, N_{it} is the number of the most relevant exemplars to $\mathbf{x}_t$ that belong to class q_i , σ is a scaling factor.

Experimental results in [74] for the English EPPS large vocabulary task showed that the kernel density approach produces a consistently better result than the conventional GMM when less than 3 hours of speech training data are available.

The k-NN method is also used in [76] where the authors proposed a method for learning label embeddings to model the similarity between labels within a nearest neighbor framework. These estimates were then applied to acoustic modeling for speech recognition. Experimental results showed that significant improvements in terms of word error rate (WER) are achieved on a lecture recognition task over a state-of-the-art baseline GMM model.

One major issue in an exemplar-based system for speech recognition is its high computational cost during decoding. To address this, in [75], two methods were proposed. The first method was proposed to quickly select a subset of informative exemplars among millions of training examples. The second method makes approximations to the sparse representation computation such that a matrix-matrix multiplication is reduced to a matrix-vector product. Experiments conducted in four large vocabulary tasks indicated improvements in speedup by a factor of four relative to the original method as well as improvements in WER in combination with a baseline HMM system.

Our recent work [31] is the first study that applies the exemplar-based approach for cross-lingual acoustic modeling. Specifically, cross-lingual bottleneck feature generated by a source language bottleneck DNN is used as the input feature for an exemplar-based acoustic model of the target language. Chapter 5 of the thesis will present the proposed cross-lingual exemplar-based acoustic model in details.

2.3 Summary

In this chapter, a conventional speech recognition system is first presented followed by a review of the state-of-the-art cross-lingual acoustic modeling techniques. Among these techniques, the cross-lingual phone mapping and the exemplar-based approaches are ones

of this thesis’s focuses. In next three chapters, the proposed methods will be presented. Chapter 3 presents the context-dependent phone mapping approach. Chapter 4 examines the use of deep neural networks in the cross-lingual phone mapping. Chapter 5 proposes an exemplar-based acoustic model for limited training data conditions.

Chapter 3

Cross-lingual Phone Mapping

This chapter presents a novel phone mapping technique for large vocabulary automatic speech recognition of under-resourced languages by leveraging well-trained acoustic models of other languages (source languages). This strategy is based on the linguistic knowledge that speech units such as phones in different languages may be similar and hence acoustic information of some well-trained models can be mapped to a new target language. However, phones for different languages would not be identical and hence “hard mapping” may not be appropriate. If the target speech unit has an overlapping distribution with speech units from the source language, then a “soft mapping” may be more appropriate.

In the first part of the chapter, a cross-lingual linear combination method is proposed. Specifically, the target language’s phone model is generated as a weighted sum of the source language’s phone models optimized by the maximum likelihood criterion. In the second part, a nonlinear phone mapping architecture is introduced to improve performance. The nonlinear mapping is used to convert likelihood or posterior scores generated by the source acoustic model to target phone posteriors. Unlike previous studies which used context-independent phone mapping strategy, the thesis proposes a novel extension to use context-dependent triphone states as the acoustic units to achieve higher acoustic resolution.

The results in this chapter have been published in: IEICE Transactions on Information and Systems [26], IALP 2012 [27], and ISCSLP 2012 [28].

The chapter is organized as follows: Section 3.1 presents the detailed setup of cross-lingual phone mapping and databases to be used in the chapter. In Section 3.2, the

cross-lingual linear acoustic model combination method is described. In Section 3.3, a discussion of context-dependent nonlinear phone mapping is presented.

3.1 Tasks and Databases

Databases: To evaluate the performance of the proposed cross-lingual phone mapping method, the Malay language is used as the source language and the English language (WSJ0 task¹) used as the target under-resourced language. The WSJ0 English corpus has been chosen as the under-resourced language as the effect of sufficient training data for it is well known. Hence we can focus on the proposed work under the reduced training corpus. In addition, Hungarian is also used as the source language to verify the effectiveness of multiple source languages.

In the WSJ0 corpus, there are 7138 clean training sentences, or roughly 15 hours of speech data. To simulate under-resourced conditions, sentences from the 7138 sentences are randomly selected to generate smaller training sets. For testing, the clean test set is used consisting of 166 sentences, or 20 minutes of speech. In this work, the focus is on acoustic model training and it is assumed that the language model and pronunciation dictionary of the target language are available. In addition, the source language acoustic model is well trained using 100 hours of the Malay read speech corpus [81].

Language model and dictionary: The standard Wall Street Journal English bi-gram language model is used in word recognition experiments. The test set contains a vocabulary of 5k words. The CMU dictionary is used which consists of 40 phones, including the silence phone.

Features: The features used in this study are the conventional 12th-order Mel frequency cepstral coefficients (MFCCs) and C0 energy, along with their first and second temporal derivatives. The frame length is 25ms and the shift is 10ms. To reduce recording mismatch between the source and target corpora, utterance-based mean and variance normalization (MVN) is applied to both training features of Malay and training and testing features of English.

¹The Aurora-4 corpus with clean training and test setting is actually used in this study. The Aurora-4 clean data are a filtered version of the WSJ0 SI84 training data and Nov92 test data. The Aurora 4 corpus with 16kHz sampling rate version is used.

3.2 Cross-lingual Linear Acoustic Model Combination

This section first discusses the use of one-to-one phone mapping (hard-mapping) to generate an acoustic model of a desired target language from a well-trained acoustic model of another language denoted here as the source language. The one-to-one phone mapping performance is then improved using adaptation techniques. After that, the proposed method based on many-to-one phone mapping is presented. In the proposed method, the model of a target speech unit is built as a linear combination of models from the source speech units.

3.2.1 One-to-one phone mapping

To perform one-to-one phone mapping, two approaches can be used: knowledge-based [82] and data-driven methods [8].

In the knowledge-based approach [82], a human expert defines the mapping between phone set of the two languages by selecting the closest IPA counterpart. Hence, no training data is required in this method. However, one difficulty of this approach is that expert knowledge is required to formulate the linguistic relationships between languages. Although linguistic experts might in theory be able to define relations according to phonetic dictionaries, the IPA system suffers from inconsistencies caused by the principles of phonological contrast and is thus error prone [1]. To overcome this challenge, this section focuses on the data-driven approach to perform one-to-one phone mapping from Malay to English.

When training data are available, a data-driven approach [8] to find the phone mapping between the two languages is an option. In our case, the English target language phone recognizer trained with limited English training data is first used to perform forced alignment on the English training utterances. These English training utterances are then decoded by the source Malay language acoustic model into Malay phone sequences. To find the relationship between the English and Malay phones, a confusion matrix between the two phone sets is computed by a frame-wise comparison of the alignments, normalized by the summed frequency of the hypothesized phone [8]. The one-to-one phone mapping

Table 3.1: One-to-one phone mapping from Malay to English generated by the data-driven method using 7 minutes of English training data.

No	English phone (Target language)	Malay phone (Source language)	No	English phone (Target language)	Malay phone (Source language)
1	ah	@	21	l	o
2	aa	aw	22	m	m
3	ae	a	23	n	n
4	ao	o	24	ng	ng
5	aw	aw	25	ow	o
6	ay	aj	26	oy	w
7	b	b	27	p	p
8	ch	ts	28	r	r
9	d	d	29	s	s
10	dh	t	30	sh	ss
11	eh	a	31	zh	z
12	er	r	32	t	ts
13	ey	e	33	th	f
14	f	f	34	uh	j
15	g	g	35	uw	u
16	hh	h	36	v	v
17	ih	e	37	w	w
18	iy	i	38	y	nj
19	jh	dz	39	z	z
20	k	k	40	sil	sil

is then derived by picking the hypothesized phone which has the highest normalized confusion score.

Table 3.1 shows the one-to-one phone mapping from Malay to English using data-driven method with 7 minutes of English training data. Since the number of phones in English is 40 while Malay has only 34 phones, some of the English phones are mapped to the same Malay phone. There are several surprising phone mappings such as English phone *dh* is mapped to Malay phone *t* or English phone *uh* is mapped to Malay phone *j*. There are two main reasons for these poor mappings. Firstly, the performance of the English phone recognizer is poor due to it is trained using only 7 minutes of English data. Secondly, the English phones such as *dh* and *uh* are not present in Malay and hence they are poorly recognized and result in mapping errors. To address this problem, in [57], linguistic knowledge can be applied to refine the initial mapping generated by the

Table 3.2: Word error rate (WER) (%) of the one-to-one phone mapping for cross-lingual acoustic models.

Method	Amount of adaptation data (minutes)		
	0	1	7
One-to-one mapping	69.3	-	-
MLLR adaptation			
1-class-MLLR	-	65.8	64.0
8-class-MLLR	-	-	54.3
16-class-MLLR	-	-	52.2
32-class-MLLR	-	-	50.9
64-class-MLLR	-	-	50.1

data-driven phone mapping. This approach however only results in a small improvement in cross-lingual phone recognition.

Given the one-to-one phone mapping, an initial acoustic model for each phone in the target language can be obtained by copying the corresponding phone acoustic model of the source language. As illustrated in the first row of Table 3.2, for the no adaptation data condition, the one-to-one phone mapping achieves the poor word error rate (WER) of 69.3%.

If target language training data are available, they can be used to adapt the initial target acoustic model. In our experiment, maximum likelihood linear regression (MLLR) [83] is applied for adaptation. The second row of Table 3.2 shows the WER achieved when 1 and 7 minutes of adaptation data are used. Comparing to the result in the first row, an improvement in WER is observed. However, using 7 minutes of adaptation data just provides a small improvement over the case of using 1 minute. To benefit from larger amounts of target adaptation data, multi-class MLLR is applied [83]. The results for multi-class MLLR using 7 minutes of adaptation data are shown in the last four rows in Table 3.2. Significant improvements are observed as compared to the case of the single-class MLLR in the second row.

3.2.2 Many-to-one phone mapping

As discussed in the previous section, the one-to-one phone mapping can be used to build an acoustic model for the target language by copying phone models from a well-trained source acoustic model. This approach however results in a high word error rate

(WER) even when multi-class MLLR adaptation is applied. For comparison, monolingual HMM/GMM acoustic models are also built from scratch using 7 minutes of English training data. The WERs for this setup are 31.8% and 30.9% for the monophone and triphone models, respectively. These results are even better than the one-to-one phone mapping with adaptation in Table 3.2.

Fig. 3.1 presents an illustration on the differences between feature distributions of three HMM states for the same IPA phone across different languages [3]. It can be observed that the variation encountered in the feature distributions for phones with the same IPA, can be significantly different across languages. These differences are caused by a complicated product of speaker and pronunciation variation, in conjunction with a language specific component which occurs because of phonotactic differences across languages. Hence, it is not appropriate if we try to find a “hard mapping” between phone sets of different languages for cross-lingual acoustic modeling.

Figure 3.1: Differences between feature distributions of three HMM states for a same IPA phone across three languages [3].

To improve performance, one speech unit in the target language can be modeled by a combination of several speech units of the source language. In the other words, “soft mapping” by interpolating a number of sounds in the source language for a sound in the target language may be a better option. In this section, a “soft phone mapping” method is proposed for cross-lingual ASR. Specifically, as shown in Eq. (3.1), the target language

phone model is formed as a linear combination of the source language phone models,

$$p(\mathbf{o}_t | s_j^T) = \sum_{i=1}^{N_S} c_{ij} p(\mathbf{o}_t | s_i^S) \quad (3.1)$$

where $\mathbf{o}_t$ is the input feature vector of the target language, s_i^S is the i^{th} phone in the source language, s_j^T is the j^{th} phone in the target language, N_S is the number of phones in the source language, c_{ij} is the combination weight for phone pair (s_i^S, s_j^T) which satisfies the constraints:

$$\sum_{i=1}^{N_S} c_{ij} = 1; \forall j \quad (3.2)$$

$$c_{ij} \geq 0; \forall i, j. \quad (3.3)$$

In Eq. (3.1), $p(\mathbf{o}_t | s_i^S)$ can be modeled by any type of source acoustic models such as HMM/GMM or HMM/MLP. In this study, the conventional HMM/GMM is used as the source acoustic model and hence Eq. (3.1) becomes:

$$p(\mathbf{o}_t | s_j^T) = \sum_{i=1}^{N_S} c_{ij} \sum_{k=1}^{M_i^S} w_{ik}^S g(\mathbf{o}_t | \boldsymbol{\mu}_{ik}^S, \boldsymbol{\Sigma}_{ik}^S) \quad (3.4)$$

where M_i^S , w_{ik}^S , $\boldsymbol{\mu}_{ik}^S$, $\boldsymbol{\Sigma}_{ik}^S$ are parameters of the source HMM/GMM acoustic model. M_i^S is the number of Gaussian mixtures for phone s_i^S , w_{ik}^S is the mixture weight, $\boldsymbol{\mu}_{ik}^S$ is the mean vector and $\boldsymbol{\Sigma}_{ik}^S$ is the covariance matrix of the k^{th} Gaussian function $g(\mathbf{o}_t | \boldsymbol{\mu}_{ik}^S, \boldsymbol{\Sigma}_{ik}^S)$.

In the training process of the many-to-one phone mapping, only the combination weight c_{ij} is estimated using the expectation maximization (EM) algorithm [35], the other parameters M_i^S , w_{ik}^S , $\boldsymbol{\mu}_{ik}^S$, $\boldsymbol{\Sigma}_{ik}^S$ given by the source acoustic model are not modified.

Table 3.3 shows the WER of the proposed many-to-one phone mapping method. In the first experiment using 7 minutes of target training data, 40 phones in the target English are mapped from 34 phones in Malay using Eq. (3.4). With this “soft-mapping” technique, 53.3% WER is obtained. This result is much better than 69.3% WER obtained by the “hard-mapping” approach without adaptation, and only slightly worse than the systems that use multi-class MLLR adaptation in Table 3.2. However the performance is still poorer than the monolingual model.

Fig. 3.2 illustrates the combination weight matrix, $\{c_{ij}\}$ for this mapping where the x -axis is the English phones, the y -axis is the Malay phones. It can be seen that the target English phones have high value of $\{c_{ij}\}$ with not one but several source Malay phones. This re-confirms the hypothesis that it is more appropriate to use the many-to-one scheme for phone mapping. Fig. 3.3 shows the combination weights for two English phones *sh* and *th*. The results show that phone *sh* has a strong relevance with Malay phone *ss*, but phone *th* has no clear relevance with any source Malay phone. One explanation for this phenomenon is that English consonant *th* has no overlap with any phone in the Malay phone set² and hence several Malay phones are utilized to perform this mapping.

Figure 3.2: Combination weights for the proposed many-to-one phone mapping.

To improve the performance of the many-to-one phone mapping, the resolution of the mapping input and output is increased from phones to states. HMM-states are used to replace phones as the speech units for mapping. Increasing acoustic unit resolution is a common method in ASR where 3-state-HMM is used to model a phone. As shown in the

²Malay phonology http://en.wikipedia.org/wiki/Malay_phonology

Figure 3.3: Combination weights for phone “*sh*” and “*th*” in English.

Table 3.3: Word error rate (WER) (%) of many-to-one phone mapping for cross-lingual acoustic models with 7 minutes of target training data.

No	Mapping	#Source units	#Target units	WER (%)
Context-independent mapping				
1	Monophone to Monophone	34	40	53.5
2	Monophone state to Monophone	102 (34x3)	40	47.9
3	Monophone state to Monophone state	102 (34x3)	120 (40x3)	39.7
Context-dependent mapping				
4	Monophone state to Triphone state	102	243	36.3
5	Triphone state to Triphone state	498	243	27.4
6	Triphone state to Triphone state	1010	243	26.5
7	Triphone state to Triphone state	1592	243	26.3
8	Triphone state to Triphone state	5000	243	26.8

second and third rows of Table 3.3, using HMM-states as the speech units for mapping gives a significant improvement over using phones as in Experiment 1. To get even higher resolution for mapping, triphone tied-states are proposed to be used as the speech unit for mapping. Specifically, each target triphone tied-state model is formed as a linear combination of the source triphone tied-state models. While the number of tied-states of the target acoustic model is kept to be 243 due to the limited 7 minutes of English training data, the number of tied-states in the source acoustic model is allowed to vary from 102 to 5000 in the experiments of the last five rows in Table 3.3.

As shown in rows 4-8 of Table 3.3, using context-dependent triphone tied-states in

both the source and target language acoustic models can improve the mapping performance significantly. Using 1592 tied states in the source acoustic model achieves the best WER at 26.3%, this result is better than 30.9% WER obtained by the monolingual model and better than 50.1% WER obtained by the one-to-one phone mapping with multi-class MLLR in Table 3.2. However, a small increase in WER is observed when using more states in the source language acoustic model. It can be explained that when too many source language states are used, the limited training data in the target language may not be sufficient to estimate the large number of combination weights robustly.

3.3 Cross-lingual Nonlinear Context-dependent Phone Mapping

The previous section showed that the proposed many-to-one phone mapping can achieve significant improvement over the one-to-one mapping approach. The proposed mapping is however based on linear combination. To improve performance, a more powerful nonlinear phone mapping approach is examined in this section.

Nonlinear phone mapping was first used in [25]. In this approach, the MLP source language acoustic model is used to generate phone posterior probabilities for the target language speech data. These posteriors are then mapped to the phone posteriors of the target language by a nonlinear mapping, e.g. an MLP. Experimental results showed promising improvements for the target language speech recognition under limited training data conditions.

Applying the idea of nonlinear phone mapping in [25], this section proposes a novel phone mapping method called cross-lingual nonlinear context-dependent phone mapping. The phone mapping framework in [25] is extended with three major improvements.

The first improvement is to use the sharper acoustic resolution by mapping triphone states as opposed to only mapping monophone states. The use of triphone mapping has been shown to be more advantageous over monophone mapping in Section 3.2.

The second improvement examines the use of source language’s likelihood scores generated by a conventional HMM/GMM model for the mapping as opposed to posterior scores generated by a hybrid HMM/MLP model. The use of likelihood scores easily allows

Figure 3.4: Posterior-based phone mapping. N_S and N_T are the numbers of phones in the source and target languages, respectively.

different types of source acoustic models to be utilized. For example, if the HMM/GMM model is used as the source language model, many acoustic modeling techniques, such as model adaptation, can be easily applied as opposed to the source language hybrid model.

Thirdly, the use of multiple source acoustic models in the cross-lingual phone mapping framework is examined. In one scenario, two source models trained from the same source language are used together to generate scores for cross-lingual phone mapping. In another scenario, two source models from two different source languages are used. In both cases, significant improvements to word error rate (WER) are achieved.

The following subsections are organized as follows. In Subsection 3.3.1, the description of context-independent phone mapping in [25] is presented. In Subsection 3.3.2, the proposed context-dependent phone mapping is described in details. Subsection 3.3.3 discusses about combination of different types of input features for phone mapping. Subsection 3.3.4 presents the experimental setup, results, and discussions.

3.3.1 Context-independent phone mapping

This section describes the posterior-based context-independent cross-lingual phone mapping proposed in [25]. This method is shown in Fig. 3.4.

The first process is to use the source language acoustic model to convert the target language speech data to the source language phone posteriors $p(s_i | \mathbf{o}_t)$ for each target

speech feature frame $\mathbf{o}_t$ and each source language phone s_i . The source language posterior vector at time t is denoted as $\mathbf{v}_t = [p(s_1|\mathbf{o}_t), \dots, p(s_{N_S}|\mathbf{o}_t)]$, where N_S is the number of phones in the source language.

In the second process of phone mapping, the representation $\mathbf{v}_t$ is then mapped to the phone posterior vector of the target language as:

$$\mathbf{u}_t = f(\mathbf{v}_t) \quad (3.5)$$

where $\mathbf{u}_t = [p(q_1|\mathbf{v}_t), \dots, p(q_{N_T}|\mathbf{v}_t)]$ is the phone posterior vector of the target language, q_j with $j \in \{1, \dots, N_T\}$ is the j^{th} phone of the target language, N_T is the number of target phones, $f(\cdot)$ is a mapping function to be learned.

In the previous study of cross-lingual phone mapping [25], monophones or monophone states are used as the class units in both the source and target language acoustic models. As a result, the acoustic resolution of the system is low and hence limits the performance of the cross-lingual system. The experimental results in Section 3.2 showed that significant improvements can be achieved by using context-dependent triphone tied-states instead of monophones as the speech units for the proposed linear phone mapping.

In the next subsection, a context-dependent phone mapping method is proposed. In addition, multi-stream phone mapping using different source language acoustic models is also investigated. Unlike the technique proposed in Section 3.2 where the Expectation-Maximization (EM) algorithm is used to estimate the linear combination weights, in this study, a neural network, a nonlinear mapping is used to discriminatively map source acoustic scores to the target speech units. The detailed explanation will be presented in the following subsections.

3.3.2 Context-dependent phone mapping

To build a high resolution acoustic model for the target language, the input representation of the acoustic space should be as detailed as possible. It is well known that monophones are just a coarse representation of the acoustic space. In comparison, a triphone acoustic model has sharper acoustic resolution and has been widely used in HMM/GMM-based LVCSR systems [6, 84, 85]. Following this, the proposal to extend monophone mapping to triphone mapping between the source and target languages is

examined. There are several advantages of using triphone states for phone mapping. For instance, one obvious advantage is that the mainstream acoustic modeling technology for LVCSR is based on triphone modeling and hence well-trained triphone acoustic models of many popular languages can be easily obtained. In addition, in this study, we propose to use HMM/GMM as opposed to hybrid HMM/MLP architecture as in [25] for the source language acoustic modeling. With this approach, many techniques such as model adaptation for robust performance can be applied. Hence, cross-lingual phone mapping may potentially benefit from these existing techniques.

In the proposed context-dependent cross-lingual phone mapping, a target language feature frame $\mathbf{o}_t$ is encoded into a vector of source language acoustic scores, $\mathbf{v}_t$, which can be source likelihoods generated by source language HMM/GMM:

$$\mathbf{v}_t = [p(\mathbf{o}_t|s_1), \dots, p(\mathbf{o}_t|s_{N_S})] \quad (3.6)$$

or source posteriors generated by source language HMM/MLP:

$$\mathbf{v}_t = [p(s_1|\mathbf{o}_t), \dots, p(s_{N_S}|\mathbf{o}_t)] \quad (3.7)$$

where N_S is the number of tied-states in the source language acoustic model, s_i in Eq. (3.6), (3.7) is the i^{th} tied-state in the source language acoustic model. Similar to the monophone mapping, the source language triphone acoustic scores $\mathbf{v}_t$ are mapped to the target language triphone tied-state posterior vector $\mathbf{u}_t$ using Eq. (3.5) where $\mathbf{u}_t = [p(q_1|\mathbf{v}_t), \dots, p(q_{N_T}|\mathbf{v}_t)]$, q_j with $j \in \{1, \dots, N_T\}$ is now the j^{th} tied-state of the target language acoustic model, N_T is the number of target tied-states, $f(\cdot)$ is a mapping function to be learned. In this study, the mapping function f is implemented by a 3-layer MLP. The choice of the 3-layer architecture will be discussed in Section 3.3.4.5.

The training process of the proposed cross-lingual triphone mapping is illustrated in Fig. 3.5 and summarized in the following steps:

Step 1 Build the target language's conventional HMM/GMM acoustic model from the limited target training data. Use decision tree to tie the triphone states to a predefined number. Generate the triphone state label for each frame of the training data using forced alignment.

Figure 3.5: A diagram of the training process for the context-dependent cross-lingual phone-mapping.

Step 2 Evaluate feature vector $\mathbf{o}_t$ of the target language training data on the source acoustic model to generate the acoustic score vector $\mathbf{v}_t$ as in Eq. (3.6) or Eq. (3.7).

Step 3 Train the mapping MLP, $f(\cdot)$ of Eq. (3.5). Use $\mathbf{v}_t$ as the input and the triphone state label generated in Step 1 as the target class of the mapping.

Fig. 3.6 illustrates the decoding process with a cross-lingual phone mapping acoustic model for LVCSR. The steps are as follows:

Step 1 Generate the source acoustic score vector $\mathbf{v}_t$ for each input frame $\mathbf{o}_t$ of the test utterances as in Step 2 of the training procedure.

Step 2 Use the trained phone mapping $f(\cdot)$ to map $\mathbf{v}_t$ to the target language tied-state posterior vector $\mathbf{u}_t$.

Step 3 Convert the target tied-state posterior vector to scaled likelihood vector $\mathbf{l}_t = [p(\mathbf{v}_t|q_1), \dots, p(\mathbf{v}_t|q_{N_T})]$ by normalizing them with their corresponding priors $[p(q_1), \dots, p(q_{N_T})]$. The priors are obtained from the target training label.

Step 4 Use the scaled state likelihoods, together with the language model and lexicon of the target language for decoding.

Figure 3.6: A diagram of the decoding process for the context-dependent cross-lingual phone mapping.

3.3.3 Combination of different input types for phone mapping

As shown in Section 3.3.2, the proposed phone mapping can handle different types of source acoustic models and different source languages to generate acoustic scores, and hence an improvement may be obtained by combining these input streams if they provide complementary information. In this study, two levels of combination are investigated: feature combination and probability combination [5, 86–88].

The feature combination scheme is a simple and straightforward approach. As shown in Fig. 3.7, in this approach, different cross-lingual features are concatenated to form the input of the mapping. For example, in the case when we use two types of source acoustic models, e.g. HMM/GMM and HMM/MLP, likelihood scores generated by the source HMM/GMM and posterior scores generated by the source HMM/MLP models are concatenated to form the input of the mapping.

The probability combination method is illustrated in Fig. 3.8. In this study, multiple target language state posterior vectors are combined at the target language probability level using the unweighted sum rule [5]. In the case of two streams are used, combined posterior probability vector $\mathbf{u}_t^C = [p(q_1|\mathbf{v}_t^1, \mathbf{v}_t^2), \dots, p(q_{N_T}|\mathbf{v}_t^1, \mathbf{v}_t^2)]$ of the target

Figure 3.7: Feature combination in the cross-lingual phone mapping where N_{S1}, N_{S2} are the numbers of tied-states in the source language acoustic model 1 and 2, respectively.

Figure 3.8: Probability combination in the cross-lingual phone mapping.

language is computed by taking the average value of the two target state posterior vectors $\mathbf{u}_t^1 = [p(q_1|\mathbf{v}_t^1), \dots, p(q_{N_T}|\mathbf{v}_t^1)]$ and $\mathbf{u}_t^2 = [p(q_1|\mathbf{v}_t^2), \dots, p(q_{N_T}|\mathbf{v}_t^2)]$ estimated by the two phone mappings as illustrated in Eq. (3.8),

$$\mathbf{u}_t^C = \frac{\mathbf{u}_t^1 + \mathbf{u}_t^2}{2}. \quad (3.8)$$

3.3.4 Experiments

3.3.4.1 Experimental setup

Tasks: To verify the performance the proposed cross-lingual phone mapping method, similar to Section 3.2, Malay is used as the source language and English as the presumed

under-resourced language. Sentences are randomly selected from the 15 hours of the clean training set in WSJ0 to generate the training sets of sizes of 7 minutes, 16 minutes, and 55 minutes. In addition, Hungarian is also used as a source language to investigate the effect of multiple source languages in phone mapping.

Source acoustic models: Two different Malay source acoustic models are evaluated: conventional HMM/GMM model and hybrid HMM/MLP model. Both the models are trained from 100 hours of Malay read speech data [81]. In the HMM/GMM model, triphone HMMs are used and the triphone states are clustered to 1592 tied-states by using decision tree based clustering. The emission probability distribution of each tied-state is represented by a GMM with 32 Gaussian mixtures. The hybrid HMM/MLP model uses the same HMM structure as the HMM/GMM model, and state posterior probabilities are estimated by a 3-layer-MLP with 2000 hidden units. The hybrid HMM/MLP model uses input feature vectors concatenated from 9 frames of MFCC features. Both the HMM/GMM and the HMM/MLP source acoustic models have about 4 million free parameters. Two monophone based HMM/GMM and HMM/MLP source language acoustic models with 102 monophone states are also trained for comparison purpose.

Besides experiments with Malay source models, experiments with Hungarian HMM/MLP monophone model are also conducted.

MLP network training: To train MLP neural networks for the phone mapping model and the monolingual hybrid baseline models, the limited target training set is separated into two parts randomly. The first part is used as the training data to update network weights and contains around 90% of the training set. The rest is used as the development set to prevent the network from over-fitting. The cross-entropy criterion is used to train the network discriminatively. In all experiments, 3-layer MLPs with 500 hidden units are used. Experiments show that the performance of the phone mapping is quite stable when 500 or more hidden units are used. Although the amount of parameters in the phone mapping neural network is quite large, the use of early stopping criterion prevents overtraining effectively.

Transition probabilities in HMM model: In the cross-lingual and hybrid baseline acoustic models, for each HMM state, the probability of jumping to the next state is simply set to 0.5. The probability of remaining in the state is hence also 0.5.

3.3.4.2 Experimental organization

Section 3.3.4 conducts a relatively large number of experiments with different aspects of phone mapping. This subsection presents the outline of the experiments conducted in this part.

- *3.3.4.3 Baseline acoustic models:* This subsection describes three baseline systems. Two of the systems are based on HMM/GMM and HMM/MLP monolingual acoustic models which are trained directly from 16 minutes of English training data to examine how the performance of conventional acoustic modeling is affected by insufficient training data. The third system is a cross-lingual tandem acoustic model chosen as a competing system to compare against the proposed cross-lingual phone mapping system.
- *3.3.4.4 Cross-lingual phone mapping acoustic models:* This subsection reports the experimental results of the proposed cross-lingual phone mapping discussed in Section 3.3.2 trained on the same 16 minutes of English training data. Various phone mapping schemes are investigated e.g. using source language HMM/GMM and HMM/MLP models, using monophone and triphone mapping schemes. The result in this section will show the advantage of the proposed phone mapping approach over the monolingual and cross-lingual tandem models in Subsection 3.3.4.3.
- *3.3.4.5 Discussion on mapping structure:* The previous subsection uses 3-layer neural network as the phone mapping from the source language to the target language. This subsection will show that using 3-layer neural network is more advantageous over 2-layer neural network even both the two architectures use the same number of parameters.
- *3.3.4.6 Effect of training data size:* In all experiments conducted in the previous subsections, we assumed that only 16 minutes of target training data are available. This subsection examines the performance of the proposed cross-lingual phone mapping when less and more target training data are used. In this section, we also compare the experimental result of the phone mapping method with the result of the linear acoustic model combination proposed in Section 3.2.

- *3.3.4.7 Combination of different types of source language acoustic models:* Since the proposed phone mapping method can handle various types of source acoustic models to generate acoustic scores, an improvement can be obtained by combining them. This subsection conducts the combination at different levels of the two phone mappings which use the source language acoustic model HMM/GMM to generate source language likelihoods and HMM/MLP to generate source language posteriors.
- *3.3.4.8 Using multiple source languages:* This subsection investigates the performance of the proposed phone mapping method with different source languages. In addition, we show that combination of different source languages can give a better acoustic coverage and results in a significant improvement for phone mapping.

3.3.4.3 Baseline acoustic models

In this section, two baseline monolingual acoustic models for English, i.e. the HMM/GMM model and the HMM/MLP model are first described. The experiments are conducted to examine how the performance of conventional acoustic modeling is affected by insufficient training data. In addition, a cross-lingual tandem acoustic model is conducted for the comparison purpose.

a. Monolingual HMM/GMM acoustic models

Two baseline HMM/GMM acoustic models are built using 16 minutes of English training data, one is a monophone model and the other is a context-dependent triphone model. In the monophone model, there are 120 states (i.e. 40 phones x 3 states/phone); while in the triphone model, there are 243 tied-states.

Table 3.4 shows the performance of the monophone and triphone models with different model complexities. It is observed that the best triphone model (4 Gaussian mixtures / state) outperforms the best monophone model (8 Gaussian mixtures / state), although the two acoustic models contain comparable total number of parameters. It is also consistent to experiments in Section 3.2, the results show that triphone model is more robust than monophone model even when only a very limited amount of training data is available. The best WER obtained by triphone model is 23.1%. We also built a triphone HMM/GMM model with the full 15 hours of English training data and achieved 7.9%

Table 3.4: Word error rate (WER) (%) of the monolingual monophone and triphone baseline HMM/GMM models for 16 minutes of training data with different model complexities.

Number of Gaussian mixtures / state	Monophone Model	Triphone Model
2	36.2	26.6
4	29.9	<u>23.1</u>
8	<u>24.9</u>	24.2
16	25.0	-

WER. This shows that conventional HMM/GMM system does not perform well under very small training size scenarios.

b. Monolingual hybrid HMM/MLP acoustic models

Two English monolingual hybrid HMM/MLP models [64] are also trained using the same 16 minutes of training data to compare against the two HMM/GMM models. Hybrid HMM/MLP acoustic models offer several advantages over the HMM/GMM approach such as: MLPs are discriminative as compared to GMMs. Furthermore, HMM/MLP does not make parametric assumption about the distribution of inputs. The HMM/MLP approach has been applied successfully for phone recognition [89] and recently for word recognition [90].

In this experiment, MLPs are used to predict the posterior probabilities of the monophone states and triphone tied-states. First, 1 frame of MFCC is used to form the input for the MLP. The frame-level state labels used for MLP training are obtained from the above HMM/GMM baseline models. The WER for the hybrid monophone and triphone models are 22.8% and 20.5% (the second row of Table 3.5), respectively. These results show a significant improvement over the best corresponding HMM/GMM models.

Now, more frames are used as the input for MLP as suggested in recent studies in hybrid systems [37–41]. Specifically, 9 frames of MFCC are used to form the input for the MLP. Surprisingly, with this configuration, we obtain a worse result, 24.6% WER for the monophone model and 22.5% WER for the triphone model. It may be due to that using more input frames can cause over-fitting for the MLP when an extremely small training size is used. Our Interspeech 2013 paper [30] also confirmed that using single frame as the input for the hybrid HMM/MLP model is a better choice when less than 55 minutes of training data are used.

Figure 3.9: Cross-lingual tandem system for Malay (source language) and English (target language).

c. Cross-lingual tandem baseline

In this study, cross-lingual tandem systems, which were proposed for under-resourced acoustic modeling, are also investigated. In the cross-lingual tandem approach [13–15], the source MLP acoustic model is used to generate phone or state posterior scores for the target speech. These scores are then used as the feature for the target language HMM/GMM.

As shown in Fig. 3.9, the Malay MLP is used to generate state posterior probability vector v_t . The natural logarithm is applied on the posteriors to make them closer to the Gaussian distribution. As the dimensionality of posterior vector v_t is usually high, principal component analysis (PCA) is used to project the log posterior vectors to 39-dimensional feature vectors v'_t . After that, v'_t is augmented with 39-dimensional MFCCs to form 78-dimensional vectors v''_t which are used as the input feature for a target language HMM/GMM model.

In this study, two types of source language MLPs are used, i.e. monophone and triphone networks which have 102 and 1592 outputs, respectively. The results of the tandem approach with the two different types of source MLPs are shown in the third and forth rows of Table 3.5. It can be seen that both the two cross-lingual tandem models outperform the monolingual HMM/GMM and HMM/MLP models (the first two rows). These results demonstrate the benefit of using acoustic scores of a well-trained source acoustic model. However, using the source triphone MLP to generate tandem feature resulted in slightly worse WER than using the source monophone MLP. This can be explained as

follows: although feature generated by the triphone MLP may contain richer information with higher resolution, it loses more information after the dimensionality reduction step from 1592 to 39 dimensions. In the case of under-resourced language, it is hard to increase the number of preserved dimensions because of the curse of dimensionality in the target HMM/GMM model with a limited amount of training data. This result is also consistent with the recent research in [91] where the tandem approach was applied in a monolingual model. Using monophone states as the output representation of MLP is adequate to generate tandem features. Increasing the number of outputs, i.e. using tied-state triphone, generally does not help to improve even sometime it performs worse than using the monophone MLP.

Table 3.5: The WER (%) of different monolingual and cross-lingual acoustic models with 16 minutes of English training data.

No	Method	Target model	
		Monophone ($N_T = 120$)	Triphone ($N_T = 243$)
Baseline monolingual acoustic model			
1	Monolingual HMM/GMM	24.9	23.1
2	Monolingual HMM/MLP	22.8	20.5
Baseline cross-lingual tandem acoustic model			
3	Source monophone	22.2	19.4
4	Source triphone	22.9	20.1
Proposed cross-lingual acoustic model (source HMM/GMM)			
5	Source monophone ($N_S = 102$)	20.6	18.3
6	Source triphone ($N_S = 1592$)	19.3	<u>16.7</u>
Proposed cross-lingual acoustic model (source HMM/MLP)			
7	Source monophone ($N_S = 102$)	19.6	17.6
8	Source triphone ($N_S = 1592$)	18.3	<u>16.4</u>

3.3.4.4 Cross-lingual phone mapping acoustic models

Now the experiments of the proposed cross-lingual acoustic model discussed in Section 3.3.2 are reported. The model is trained on the same 16 minutes of training data as that used in the baseline models. As shown in Fig. 3.6, 39-dimensional MFCC feature vector, $\mathbf{o}_t$ is passed through the source acoustic model to obtain N_S likelihood scores from the

source HMM/GMM model or N_S posteriors from the HMM/MLP model. In this study, both context-independent and context-dependent source acoustic models are examined:

- (i) The Malay monophone acoustic model with 102 states.
- (ii) The Malay triphone acoustic model with 1592 tied-states.

These N_S scores are mapped to N_T states of the target language. N_T can be 120 states for the English monophone model or 243 tied-states for the English triphone model.

The last four rows of Table 3.5 represent the results for the proposed cross-lingual acoustic models which use the HMM/GMM and hybrid HMM/MLP source models. From the table, four major observations can be seen.

First, all proposed cross-lingual phone mappings outperform the monolingual baseline models significantly. The WERs obtained by the proposed method are also considerably better than the cross-lingual tandem approach although both approaches use source acoustic scores as the input feature. This shows that the phone mapping approach is more advantageous than the tandem approach which models the source acoustic scores by Gaussian distributions.

Second, by comparing the last four rows of Table 3.5, it is clear that using source triphone as the input of the cross-lingual phone mapping produces better results than using source monophone. This is due to the fact that the source triphone states provide more detailed representation of the target speech than the source monophone states. This result is contrary to the result on the cross-lingual tandem approach (row 3, 4). It can be explained that MLP phone mapping can handle all inputs and does not lose information via the dimensionality reduction step as in the tandem approach and hence can take advantage of higher resolution feature generated by the source language context-dependent model. This is explored further in Section 3.3.4.5.

Third, by comparing the last two columns of the table, it is observed that using target language triphone states as the label of the phone mapping consistently outperforms using target language monophone states. The best performance of the cross-lingual phone mapping is WER=16.7% and 16.4% for the source HMM/GMM and source HMM/MLP, respectively. These results are obtained by using triphone representation in both the source and target language acoustic models.

Fourth, using the source hybrid HMM/MLP can produce a small improvement over the source conventional HMM/GMM. However, with the best configuration (i.e. triphone-to-triphone mapping), the performance of the two systems is almost the same.

In summary, the results in Table 3.5 show that the cross-lingual phone mapping outperforms the monolingual acoustic models as well as the cross-lingual tandem model. In addition, the proposed context-dependent cross-lingual phone mapping produces significantly better results than the context-independent cross-lingual phone mapping in [25].

3.3.4.5 Discussion on mapping structure

The previous subsection showed that for the case of limited training data, the proposed phone mapping method using MLPs is much more effective than the tandem approach which tries to model source acoustic scores using Gaussian distributions. It demonstrates that choosing an appropriate model to relate the source acoustic scores to target phone states is important. In the experiments discussed in Subsection 3.3.4.4, 3-layer MLPs were used as the phone mapping from the source language to the target language. In this subsection, the reason for selecting this MLP topology is presented.

As stated in the previous subsections, source acoustic scores, i.e. posteriors, likelihoods can be considered as higher level features as compared to conventional features such as MFCCs for speech recognition. This raises the question of whether we can use a simpler mapping to perform this task. To answer this question, experiments are conducted using 2-layer neural networks (NN) (i.e. with no hidden layer). In the case of monolingual hybrid HMM/MLP, the MLP can be considered as a mapping from the input cepstral feature, i.e. MFCCs to HMM states.

Table 3.6 shows the phone mapping results with different inputs and different NN architectures for 16 minutes of target English training data. Three different types of input are:

- MFCCs with 39 inputs (i.e. 1 frame of 39-dimensional MFCC vector)
- Source monophone likelihoods with 102 inputs
- Source triphone likelihoods with 1592 inputs

Table 3.6: The WER (%) of different mapping architectures with 16 minutes of target English training data (the target acoustic model is triphone). Number in (.) in the first and the last column indicates the number of inputs and number of hidden units, respectively. Number in (.) of the third column represents relative improvement over the corresponding 2-layer NN in the second column.

Input	Mapping architecture		
	2-layer NN	3-layer NN (500HUs)	3-layer NN (*)
MFCC (39)	30.6	20.5 (33.0%)	24.5 (34HUs)
Source monophone (102)	21.6	18.3 (15.3%)	20.3 (72HUs)
Source triphone (1592)	17.6	16.7 (5.0%)	16.9 (211HUs)

243 outputs are used to model 243 tied states in the target context-dependent acoustic model. The second column represents WER of using 2-layer NN. The third column is the result using 3-layer NN with 500 neurons in the hidden layer presented in the previous section. The number in (.) represents relative improvement over 2-layer NN mapping. By comparing these two columns of Table 3.6, it is observed that in all three cases, 3-layer NN outperforms corresponding 2-layer NN significantly, especially for the case of MFCC input. The results show that for low level MFCC features, 3-layer MLP performs much better than 2-layer NN due to that the system needs to be powerful enough to accurately map MFCC to states of the target language. For high level features such as source triphone scores, although the difference between the 2-layer and 3-layer networks is smaller, a more flexible 3-layer network is still preferred. This result is also consistent with the result reported in [25] where the posterior weighted product-of-expert approach realized by a 3-layer NN outperformed the product-of-posterior model realized by a 2-layer NN in a cross-lingual phone recognition task.

Now we investigate whether the improvement of 3-layer NNs over 2-layer NNs comes from larger number of parameters or their 3-layer architecture. The result shown in the last column of Table 3.6 is obtained using 3-layer NNs those have the same number of parameters as the corresponding 2-layer NNs. Number in (.) represents number of hidden units in the 3-layer NN. It is observed that although the performance deteriorates when smaller hidden layers are used, 3-layer NNs perform better than 2-layer NNs even with the same number of parameters. One advantage of using 3-layer NNs is that while it is impossible to change number of parameters in 2-layer NNs, it is very easy to select a

suitable model complexity for different phone mapping problems by changing number of hidden units in 3-layer NNs. Note that in all of mapping experiments in this study, 3-layer NNs with 500 hidden units are simply chosen. It is believed that further improvement can be obtained if this parameter is chosen carefully for each experiment. From the finding in this subsection, in all next experiments, 3-layer MLP will be used to realize phone mapping.

3.3.4.6 Effect of training data size

In all previous experiments, we assumed that only 16 minutes of target English training data are available. In this section, we examine the effect of different training data sizes on the performance of the cross-lingual phone mapping acoustic model. Three training data sizes are used in our study, i.e. 7 minutes, 16 minutes, and 55 minutes of English target data which are randomly selected from 15 hours training data of the WSJ0 corpus.

The experiments in Section 3.3.4.4 showed that using context-dependent triphone at both the source and target language acoustic models can improve the phone mapping performance significantly. Therefore, in this section, both the source and target language acoustic model are context-dependent triphone. The source language acoustic model consists of 1592 tied-states. For the target language acoustic model, with each training data size, the training steps are followed the procedure in Section 3.3.2 to build the phone mapping. The number of tied states in the target acoustic model is optimized to achieve the best performance for the monolingual HMM/GMM model. This number varies for different amounts of available target training data. Specifically, 243, 243 and 501 triphone tied states are used for the cases of 7, 16, 55 minutes of training data, respectively.

As opposed to the phone mapping model, in the cross-lingual tandem model, the source language MLP is monophone while the target language acoustic model is triphone. The reason has been shown in Section 3.3.4.3 that using source language triphone MLP results in a poor performance due to information lost during the dimensionality reduction step.

Fig. 3.10 shows the performance of five different acoustic models in WER with three different amounts of training data. The first three columns represent the performance of the three baseline models which include monolingual models: HMM/GMM, HMM/MLP

Figure 3.10: The WER (%) of different acoustic models with different amounts of target training data.

and the cross-lingual tandem model. It can be seen that the performance of all baseline models degrades quickly when less training data are used. The cross-lingual tandem approach outperforms the two monolingual models significantly for all three data sizes.

The last two columns in Fig. 3.10 show the performance of the two proposed context-dependent phone mapping systems which use the source HMM/GMM and hybrid HMM/MLP models, respectively. It is observed that the proposed systems achieve a significant improvement over the monolingual as well as the cross-lingual baseline systems especially when a small amount of target training data is available. Also note that although both the proposed phone mapping and the cross-lingual tandem approaches use source acoustic scores as the input feature, modeling them by a mixture of Gaussian distributions in the tandem approach does not work well for the case of very limited training data. In this case, finding the “mapping” between the source phone set and the target phone set as in the proposed method is more effective. We also observe that for the phone mapping, by

Figure 3.11: The WER (%) of individual and combined phone mapping models.

using likelihood scores generated by the source language HMM/GMM or posterior scores generated by the source language HMM/MLP, we can achieve a similar performance.

For the case of 7 minutes of training data, the results of the proposed cross-lingual phone mapping are better than the result of using linear acoustic model combination presented in Table 3.3 of Section 3.2.2. It shows that using MLP mapping can bring better performance due to two main reasons. The first reason is that nonlinear MLP is used as opposed to linear combination. The second reason is that the discriminative training criterion, i.e. cross-entropy is used to train the MLP mapping as opposed to the generative training criterion, i.e. maximum likelihood in the linear combination method.

3.3.4.7 Combination of different types of source language acoustic models

Since the proposed method can handle various types of source acoustic models to generate acoustic scores for phone mapping, an improvement can be obtained by combining them. In this study, the two phone mappings presented in the previous subsections which use source HMM/GMM and source HMM/MLP are combined at the feature level (Fig. 3.7) and the probability level (Fig. 3.8). The results for the individual and combined phone

mapping models are shown in the Fig. 3.11. It is clearly shown that the combined systems outperform both the individual phone mappings consistently. Although the two source acoustic models are trained with the same data from the same language, the different model structures and different training criteria make information generated by the two acoustic models complementary. While the source language HMM/GMM is trained on a maximum likelihood criterion, the source language HMM/MLP models use a discriminative criterion to optimize the model parameters.

There is also a difference between the two combination methods. Combination at the probability level gives a better performance over combination at the feature level especially when a very small amount of training data is available. It can be explained that although the feature combination approach seems to be a better choice for multi-stream input for the MLP mapping, it may suffer from overfitting when a very small amount of training data is available. Especially, when source triphone models are used, the number of inputs for the mapping is large if we concatenate multi-streams at input level. In this case, the probability combination method can be a better option.

3.3.4.8 Using multiple source languages

In all previous experiments, only Malay is used as the source language. This subsection investigates the performance of the proposed phone mapping method with different source languages. A well-trained monophone hybrid MLP Hungarian phone recognizer from Brno University of Technology (BUT)³ [92] is used. Given English speech data, the Hungarian recognizer produces 186 monophone state posterior probabilities for each speech frame. These probabilities are mapped to English tied-states as in the Malay-to-English case. Note that BUT Hungarian recognizer uses 8-kHz sampling waveform as the input while our target English corpus (i.e. WSJ0) is 16-kHz. We need to down-sample the target corpus to 8-kHz before applying the Hungarian recognizer.

Fig. 3.12 shows the results of the two phone mapping models with the two source languages. The first column is WER of the phone mapping with the source language Malay reported in the previous sections. In this case source acoustic scores are generated by the Malay triphone HMM/GMM model. The result for Hungarian-to-English mapping

³<http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context>

Figure 3.12: The WER (%) of the phone mapping model for two source languages with three different amounts of target training data.

is shown in the second column. It can be seen that when a very small amount of target training data is available (i.e. 7 and 16 minutes) using the Hungarian source acoustic model provides better performance than using the Malay model. One possible reason is that Hungarian language and English language are more similar than the Malay-English pair. As a result, the Hungarian-to-English phone mapping can be implemented easier and hence provides better performance even with an extremely small amount of target training data. However, when more training data are available, using the source Malay model achieves better performance. This can be explained as the BUT Hungarian MLP is a monophone recognizer while the Malay HMM/GMM model is a triphone model which provides higher resolution feature input for phone mapping. This will be useful when we have more target data to train the mapping.

The last two columns of Fig. 3.12 show the results when Malay-to-English and Hungarian-to-English mappings are combined at the feature and probability levels. Interestingly, the combined models provide a big improvement over both the individual mappings. It demonstrates that the two acoustic models of the two source languages provide complementary information. In other words, combination of different source lan-

guages can give a better acoustic coverage for phone mapping. It is also noted that with 55 minutes of English training data, the best combined phone mapping can give 9.0% WER which is close to 7.9% WER of the monolingual HMM/GMM model trained with the whole 15 hours of English training data.

There is also a slight difference between the two combination approaches. While probability combination outperforms feature combination for the case of 7 minutes of training data, with bigger amounts of training data, i.e. 16 and 55 minutes, feature combination is a better choice. Note that the dimension of posterior vectors generated by the Hungarian phone recognizer is only 186, which is much lower than the number of tied states in the Malay triphone model (1592 tied states). Hence, it is less likely to have overtraining problem when 16 minutes or more English training data are available. This may explain why in this case, feature combination is better than probability combination in Fig. 3.12 and vice versa in Fig. 3.11 (the last two columns).

3.4 Conclusion

This chapter presented two novel phone mapping techniques for large vocabulary automatic speech recognition of under-resourced languages by leveraging well-trained acoustic models of other languages. In the first part of the chapter, a cross-lingual linear acoustic model combination method was presented. Specifically, phone model of the target language is assumed to be a weighted sum of all phone models in the source language acoustic model. In the second part, a nonlinear phone mapping architecture was presented. In this mapping, the source acoustic model is used to generate likelihood or posterior scores, these scores are then mapped to target phone posteriors using a nonlinear mapping, e.g. MLP. Experimental results verified the effectiveness of the proposed phone mapping technique for building LVCSR models under limited training data conditions. There are two advantages in the proposed methods, i.e. the use of triphone states for improved acoustic resolution of both the source and target models and the ability to apply various types of source acoustic models which results in an additional improvement when combining them even they are trained from the same data. The experimental results indicated that using nonlinear phone mapping with discriminative training criterion

(cross-entropy) achieves better performance than using linear combination with generative training criterion (maximum likelihood). In addition, the combination of different source languages can significantly improve the performance of phone mapping as we may have a better acoustic coverage for the target language.

This chapter showed that using 3-layer neural networks as the mapping can achieve better results over simpler 2-layer mapping networks. In the next chapter, the proposed phone mapping will be investigated using deeper structures for the source and target language acoustic models.

Chapter 4

Deep Neural Networks for Cross-lingual Phone Mapping Framework

In the previous chapter, the focus was to increase the mapping resolution by using context information. In this chapter, the focus is shifted to improve the quality of cross-lingual features used for mapping by using deeper structures such as Deep Neural Networks (DNNs). DNNs proposed by Hinton *et al.* in 2006 [42] is a new powerful machine learning technique with multiple nonlinear layers. They have been used effectively for handwriting recognition [42], 3-D object recognition [93], dimensionality reduction [94] and recently in speech recognition from small [37, 39] to very large tasks [40, 41].

In this chapter, two approaches are investigated to improve the cross-lingual phone mapping using DNNs. In the first approach, DNNs are used as the source language acoustic model. This is motivated by the fact that if DNNs can model the source language better than shallow models, the source language posterior features generated from DNNs will also be better and should result in better performance of phone mapping. In the second approach, DNNs are used to replace 3-layer MLPs to realize the phone mapping function. The results presented in this chapter have been published in APSIPA ASC 2011 [29] and INTERSPEECH 2013 [30].

The chapter is organized as follows. Section 4.1 introduces DNNs and their application for monolingual speech recognition. Section 4.2 applies DNNs to improve cross-lingual phone mapping. Section 4.3 concludes the chapter.

4.1 Deep Neural Network Introduction

In this section, the concept of deep learning and DNNs is introduced in details. Subsection 4.1.1 provides an introduction to deep architectures and their advantages over shallow architectures. Subsection 4.1.2 introduces a basic block of DNNs called Restricted Boltzmann Machines (RBMs). Subsection 4.1.3 presents the applications of DNNs for speech recognition. Subsection 4.1.4 shows experimental results of using DNNs for monolingual speech recognition.

4.1.1 Deep architectures

Deep learning is a relatively new area of machine learning first proposed by Hinton *et al.* in 2006 [42]. It has been effectively used for handwriting recognition [42], 3-D object recognition [93], dimensionality reduction [94] as well as speech recognition tasks [37–41].

The concept of deep structures is defined from neural networks. A multilayer perceptron neural network (MLP) with many hidden layers is an example of deep architecture models. The complexity theory of circuits [95] strongly suggests that deep architectures are much more efficient in terms of required computational elements than shallow architectures such as hidden Markov models, neural networks with only one hidden layer, conditional random fields, kernel regression, support vector machines, and many others.

Deep architecture models have many levels of non-linearities which allow them to compactly represent highly nonlinear functions. Unfortunately, such deep networks are hard to train. In general, since models with deep architectures consist of several nonlinear layers, the associated loss functions are almost always non-convex. If the weights of the networks are initialized randomly, then the local gradient-based optimization algorithms such as backpropagation will often be trapped in poor local optima. To overcome this, Hinton *et al.* [42] introduced a moderately fast, unsupervised learning procedure for deep generative models called Deep Neural Networks (DNNs). The greedy layer-by-layer training is the key feature of this algorithm in order to efficiently learn a deep, hierarchical probabilistic model. This learning algorithm can optimize the network weights at time complexity linear to the size and depth of the network. There are several distinct characteristics of the algorithm [96]:

- The greedy layer-by-layer learning algorithm has been shown to be able to find a good set of model parameters even for very large models containing many nonlinear layers with millions of parameters.
- The algorithm can efficiently use unlabeled data for the unsupervised pre-training process. Labeled data are only used at the final step to train the network weights for classification.

The next section will introduce a basic block of DNNs called Restricted Boltzmann Machines (RBMs).

4.1.2 Restricted Boltzmann machines

A DNN is made by stacking several bipartite undirected graphical models called restricted Boltzmann machines (RBMs). A RBM is a realization of Markov random fields with a two-layer architecture as illustrated in Fig. 4.1(a). There are two types of units in the architecture, they are: visible (typically Bernoulli or Gaussian) stochastic units $\mathbf{v}$ are connected to hidden (typically Bernoulli) stochastic units $\mathbf{h}$. Normally, all the visible

Figure 4.1: A DNN(b) is composed by a stack of RBMs (a).

units are connected to all the hidden units and there is no visible to visible or hidden to hidden unit connection. The weights of the connections and the biases of the individual units form a joint probability distribution $p(\mathbf{v}, \mathbf{h}|\theta)$ over $\mathbf{v}$ and $\mathbf{h}$ given model parameters θ . This distribution is defined in terms of an energy function $E(\mathbf{v}, \mathbf{h}|\theta)$ [96] as

$$p(\mathbf{v}, \mathbf{h}|\theta) = \frac{\exp(-E(\mathbf{v}, \mathbf{h}|\theta))}{Z(\theta)} \quad (4.1)$$

where $Z(\theta)$ is known as the normalizing constant

$$Z(\theta) = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}|\theta)). \quad (4.2)$$

The marginal probability that the model assigns to a visible vector $\mathbf{v}$ is

$$p(\mathbf{v}|\theta) = \frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}|\theta))}{Z(\theta)}. \quad (4.3)$$

With different types of visible and hidden units, different energy functions are defined. In the case when both the visible and the hidden units are Bernoulli stochastic units, the energy function is defined [96] as

$$E(\mathbf{v}, \mathbf{h}|\theta) = - \sum_{i=1}^V \sum_{j=1}^H w_{ij} v_i h_j - \sum_{i=1}^V b_i v_i - \sum_{j=1}^H a_j h_j \quad (4.4)$$

where model parameters $\theta = \{\mathbf{w}, \mathbf{b}, \mathbf{a}\}$, w_{ij} is the weight between visible unit i and hidden unit j ; b_i and a_j are biases for visible unit i and hidden unit j , respectively; V , H are the number of visible units and hidden units, respectively.

Since there is no visible-visible connection, all of the visible units become independent given the hidden units, and vice versa. The conditional distributions, when both the visible and hidden units are Bernoulli stochastic units, can be effectively derived [96] as

$$p(h_j = 1|\mathbf{v}, \theta) = \sigma \left(\sum_{i=1}^V w_{ij} v_i + a_j \right) \quad (4.5)$$

$$p(v_i = 1|\mathbf{h}, \theta) = \sigma \left(\sum_{j=1}^H w_{ij} h_j + b_i \right) \quad (4.6)$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoidal function.

For the case when the visible and hidden units are Gaussian and Bernoulli stochastic units respectively, the energy function is defined as

$$E(\mathbf{v}, \mathbf{h}|\theta) = - \sum_{i=1}^V \sum_{j=1}^H w_{ij} v_i h_j + \frac{1}{2} \sum_{i=1}^V (v_i - b_i)^2 - \sum_{j=1}^H a_j h_j \quad (4.7)$$

and the conditional probabilities of hidden unit h_j and visible unit v_i are

$$p(h_j = 1|\mathbf{v}, \theta) = \sigma \left(\sum_{i=1}^V w_{ij}v_i + a_j \right) \quad (4.8)$$

$$p(v_i|\mathbf{h}, \theta) = \mathcal{N} \left(\sum_{j=1}^H w_{ij}h_j + b_i, 1 \right) \quad (4.9)$$

where v_i is a real number which follows a Gaussian distribution $\mathcal{N}(\cdot)$ with the mean $\sum_{j=1}^H w_{ij}h_j + b_i$ and unit variance.

The Gaussian-Bernoulli RBMs are normally used to convert real-valued stochastic visible units to binary stochastic variables and so that they can be further processed by other Bernoulli-Bernoulli RBMs.

Eq. (4.5), (4.8) allow the use of RBM weights to initialize an MLP with sigmoidal hidden units because we can equate the inference for RBM hidden units with forward propagation in a neural network.

The goal of learning in a RBM is to change the model parameters θ to maximize the probability of the data $p(\mathbf{v}|\theta)$. The training process can be done effectively by a procedure called “Contrastive Divergence” [97]. After training the RBMs, they are stacked to build the DNN. In the next section, the application of DNNs for speech recognition will be presented.

4.1.3 Deep neural networks for speech recognition

The recent success of DNNs for various problems has interested speech researchers to apply them for ASR tasks. Mohamed and Hinton [38] used interpolating conditional RBMs (ICRBMs) for phone recognition on the TIMIT corpus. A good accuracy was achieved using a simple one layer RBMs. In another work [37], a four-layer DNN was used to estimate the 183 state posteriors of 61 monophone for 3-state-HMM models. Although monophone models were used, the hybrid HMM/DNN system provided a competitive phone error rate on the TIMIT corpus. However, when more layers were added to the DNN, no significant improvement was observed. In [41], DNNs were combined with conditional random fields (CRFs) instead of HMMs to model the sequential information. In this research, the DNN weights as well as the state-to-state transition parameters

and language model scores were optimized by using a sequential discriminative training criterion. A higher accuracy was achieved over the HMM/DNN model with the frame discriminative training criterion on the TIMIT corpus.

In the Gaussian-Bernoulli RBM architecture, the inputs are conditionally independent given the hidden unit activations. This assumption is inappropriate for speech data when many frames are concatenated to form an input vector. To relax this assumption, in [98], mean-covariance RBMs (mcRBMs) were proposed and Dahl et al. [39] applied the mcRBM for ASR tasks and achieved a very low phone error rate of 20.5% on the TIMIT corpus.

The success of HMM/DNN on phone recognition tasks motivated speech researchers to apply deep structures to LVCSR with much larger vocabularies and more varied speaking styles. The first successful use of HMM/DNN acoustic models for a large vocabulary task used data collected from the Bing mobile voice search application. The results reported in [99] showed that using HMM/DNN acoustic model with context-dependent states achieves 69.6% sentence accuracy on the test set, compared with 63.8% for a strong, minimum phone error (MPE)-trained HMM/GMM baseline. The HMM/DNN training recipe developed for the Bing voice search data was applied to the Switchboard speech recognition task [40]. A DNN with 7 hidden layers and 2,048 units in each layer and full connectivity between adjacent layers are used to replace the GMM in the acoustic model. Experimental results showed that using DNNs reduced the word error rate to 18.5% which is significantly better than 27.4% of the baseline HMM/GMM.

Now, we discuss in details about how to build a DNN for speech recognition. Basically, as shown in Fig. 4.2, four main steps are required to train a DNN.

Step 1: The pre-training procedure is conducted for the first RBM [97], and then the activation probabilities of the first RBM's hidden units are used as the visible data for the second RBM. This process is then repeated to train upper RBMs.

Step 2: After n^{th} RBMs are trained, the final layer is added to represent the desired outputs such as HMM-states.

Step 3: The class training labels are provided for each frame in the training set. These labels can be obtained from other acoustic models such as HMM/GMM.

Figure 4.2: Four steps in training a DNN for speech recognition.

Step 4: A discriminative learning procedure such as back-propagation is used to fine-tune all the network weights jointly using the labeled training data.

In the decoding process, the DNN is treated as a conventional neural network with the same architecture. The next section will present the application of the above DNNs for two monolingual speech recognition tasks.

4.1.4 Experiments of using DNNs for monolingual speech recognition

In this section, the performance of the hybrid HMM/DNN model is investigated for monolingual tasks. Its performance is compared with shallow models such as HMM/GMM and HMM/MLP models. This section also investigates the advantages of RBM pre-training

for DNN weight initialization over the random weight initialization scheme. Two speech recognition experiments are conducted. The first experiment is phone recognition on the TIMIT database¹. The second experiment evaluates the HMM/DNN acoustic model in an LVCSR task under limited training data conditions on the WSJ0 corpus.

4.1.4.1 HMM/DNN acoustic model for phone recognition

a. Experimental setup

The phone recognition performance of different acoustic models is evaluated on the TIMIT speech database. The training set consists of 3696 utterances from 462 speakers which is around 3 hours. A small subset of speakers is extracted from the training set (50 speakers) and used as the development set. The complete test data set contains 1344 utterances from 168 speakers. No language model is used. Our APSIPA 2011 paper [29] gives the explicit experimental setup details.

b. Shallow baseline acoustic models

Before examining the HMM/DNN model, we first present the recognition results given by two conventional shallow models: HMM/GMM and HMM/MLP.

Fig. 4.3 illustrates the phone error rate (PER) on the training and test sets provided by the HMM/GMM model with different model complexities. It is observed that the PER in both the training and test sets reduces consistently when the model complexity increases from 1 to 128 Gaussian mixtures per state. Over-fitting occurs when higher complexity models are used, i.e. while the PER in the training set drops rapidly, the PER in the test set rises. This shows that with the HMM/GMM model adding more parameters does not always bring more benefit when there is insufficient amount of training data. The best PER in the test set obtained by the conventional HMM/GMM system is 33.6% when the number of Gaussian mixtures per state is 128.

Another shallow acoustic model is the hybrid HMM/MLP [64]. In this case, the MLP replaces the GMM of the HMM/GMM model to produce the state likelihood score. A 3-layer MLP with 1024 hidden units is selected. To train the MLP, each frame in the training data is assigned a state label. This label is generated using forced alignment

¹<http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1>

Figure 4.3: Phone error rate (PER) on the training and test sets of the HMM/GMM model with different model complexities.

with the above HMM/GMM model. The hybrid HMM/MLP achieves 27.3% PER which is much better than 33.6% from the HMM/GMM model.

c. Deep acoustic models

The previous part examined the two shallow acoustic models i.e. HMM/GMM and HMM/MLP. In this part, we investigate deep acoustic models to answer two questions: (i) can improvement be achieved using deeper structure of DNN? (ii) for DNN models, would RBM pre-training initialization help to improve performance over random weight initialization?. To answer these questions, experiments will be conducted to evaluate DNNs with the number of layers from 3 to 8 and with both the RBM pre-training initialization and the random weight initialization schemes. In this experiment, the number of hidden units in each hidden layer is 1024.

Fig. 4.4 presents the PER of the HMM/DNN models. The x-axis shows the number of layers i.e. how deep is the network, and the y-axis shows the phone error rate. Performances of the HMM/DNN models using the two initialization schemes are shown as the green and blue lines. Note that when 3-layer architecture with random weight initialization is used, DNN becomes conventional MLP in the previous experiment. In

Figure 4.4: Performance of the HMM/DNN acoustic model for the two initialization schemes and the combined models with different numbers of layers.

this case, the results show that even shallow network can benefit from using RBM pre-training. Adding the second hidden layer gives better performance for DNN with the two initialization schemes. However, with deeper architectures, the DNN with random initialization does not achieve further improvement, and its performance even deteriorates when number of layers is larger than 6. This shows empirically that the back propagation algorithm cannot handle well for deep structures with random weight initialization. Whereas the performance of the DNN with RBM pre-training initialization shows that using more hidden layers can improve performance of the model.

Now, we investigate the combination of the DNN models with the two initialization schemes above. Combination of information from different ASR systems generally improves speech recognition accuracy [5, 86–88]. The work in Chapter 3 (Section 3.3.4.7) showed that combination of different phone mapping systems can give a consistent improvement over the individual systems. The reason for this advantage is that different systems often provide different errors. In this experiment, the combination of the above two DNN models is examined i.e. combination of the DNN with RBM pre-training initialization and the DNN with random weight initialization. The PER of the combined

models is shown in Fig. 4.4 (red and purple lines) when the two DNN models are combined at the probability level using either the product or sum rules [29] (more details can be found in our APSIPA ASC 2011 paper [29]). Interestingly, significant improvements are achieved over both the two individual systems. Although the two models have an identical structure and training data, the different weight initialization schemes make information generated by the two DNNs is complementary.

4.1.4.2 HMM/DNN acoustic model for LVCSR under limited training data conditions

The previous subsection showed that using the HMM/DNN acoustic model achieves a significant improvement over shallow models on a phone recognition task. In this subsection, we investigate the HMM/DNN model under limited training data conditions for an LVCSR task.

a. Experimental setup

Training data are randomly extracted from the WSJ0 corpus to generate four subsets with 7, 15, 55 and 220 minutes of training data. The number of tied states is optimized to achieve the best performance for the HMM/GMM model. This number varies for different amounts of available target training data. Specifically, 243, 243, 501 and 1005 triphone tied-states are used for the cases of 7, 16, 55 and 220 minutes of training data, respectively.

Since the amount of training data is small, small-scaled neural networks with 500 units for each hidden layer are used. Four acoustic models are examined:

- HMM/GMM baseline model.
- HMM/MLP model with random weight initialization: MLP with 3-layer architecture.
- HMM/DNN model with random weight initialization: DNN with 5-layer architecture.
- HMM/DNN model with RBM pre-training initialization: DNN with 5-layer architecture.

Figure 4.5: Comparison of HMM/MLP and HMM/DNN on an LVCSR task for different training data sizes.

b. Experimental results

Fig. 4.5 shows the performance of above four models under four limited training data sizes. It is observed that the performance of all four models drops quickly when limited training data are used and three hybrid models outperform significantly the HMM/GMM baseline model. We also see that using HMM/DNN (third column) achieves a significant improvement over the two shallow models i.e. HMM/GMM (first column) and HMM/MLP (second column) even under very limited training data conditions. Now, we investigate the effect of the RBM pre-training initialization over random initialization for DNN. As shown in the fourth column, using the RBM pre-training scheme achieves a consistent improvement over random weight initialization except the case of 7 minutes of training data.

4.1.5 Conclusion

This section introduced the deep architectures and their advantages over the shallow architectures for speech recognition. The advantages of DNNs were then verified by ex-

periments conducted on a monolingual phone recognition task and a monolingual LVCSR task under limited training data conditions. The next section will present the work of applying DNNs to improve the proposed cross-lingual phone mapping method.

4.2 Deep Neural Networks for Cross-lingual Phone Mapping

In the proposed work of cross-lingual phone mapping presented in Chapter 3, both the source acoustic model and phone mapping function were implemented using shallow models, i.e. HMM/GMM, HMM/MLP were used as the source models and MLPs were used for the phone mapping function. Section 4.1.4 has shown the advantages of deep models over shallow models for monolingual speech recognition tasks. The aim of this section is to investigate the use of DNNs for cross-lingual phone mapping.

In this section, two approaches to use DNNs for the cross-lingual phone mapping are examined. The first approach is to use DNNs as the acoustic model of the source language. It is hypothesized that if DNNs can model the source language better than shallow models such as MLPs or GMMs, source language posterior feature generated from DNNs will also be better and will result in better performance for the phone mapping task. For comparison, source language bottleneck DNN [61] is also used to generate cross-lingual bottleneck feature for the target acoustic models. In the second approach, DNNs are used to replace the MLPs to realize the phone mapping function. Again, this is motivated by the fact that DNNs may have better mapping capability than shallow models such as MLPs. These two approaches result in three experimental setups:

- **Setup A:** using DNNs as the source language acoustic model to generate cross-lingual posterior feature for phone mapping.
- **Setup B:** using DNNs to extract cross-lingual bottleneck feature for target language acoustic models (used as a baseline to compare to *Setup A*).
- **Setup C:** using DNNs to implement the phone mapping function.

These three setups will be presented in the next subsection.

4.2.1 Three setups using DNNs for cross-lingual phone mapping

4.2.1.1 Setup A: using DNNs as the source language acoustic model to generate cross-lingual posterior feature for phone mapping

Section 4.1.4 showed that using DNNs for acoustic models achieves a significant improvement over shallow models for both monolingual phone recognition and word recognition. It is interesting to investigate whether the benefit of DNN-based source language acoustic models will also propagate to the target language after cross-lingual phone mapping. The using of DNN-based source acoustic models in phone mapping is straightforward and shown in Fig. 4.6. The shaded steps represent the source DNN acoustic model (called POS-DNN where POS stands for posterior feature) to generate posterior feature. The output layer presents monophone or triphone states of the source language acoustic model. The output from source DNN model is fed into the phone mapping neural network to map to target language state posteriors. If we view the phone mapping module as a conventional hybrid system, the DNN source acoustic model acts as a feature extractor that is well-trained from a large amount of data in the source language. In this approach, the cross-lingual phone mapping is able to benefit from recent progress in DNN-based acoustic modeling techniques [100]. It is expected that a sharper DNN source language acoustic model is able to describe the target language speech data better, hence making the cross-lingual phone mapping be an easier task.

4.2.1.2 Setup B: using DNNs to extract cross-lingual bottleneck feature for target language acoustic models

This setup is conducted as a baseline to compare to our phone mapping approach in *Setup A* where DNNs are used as the source language acoustic model. Using DNNs to extract bottleneck feature is also a popular way to utilize the capability of DNNs to improve speech recognition performance. There are also several studies that investigate the portability of bottleneck feature trained from one language to use as the feature for acoustic models of another language [16, 17]. This approach is illustrated in Fig. 4.7. In the bottleneck network (called BN-DNN where BN stands for bottleneck feature), there is a bottleneck layer that is usually 30 to 40 nodes. The purpose of using such bottleneck

Figure 4.6: Illustration of using DNN as the source language acoustic model to generate cross-lingual posterior feature for phone mapping (*Setup A*).

Figure 4.7: Illustration of using DNN to extract cross-lingual bottleneck feature for target language acoustic models (*Setup B*).

Figure 4.8: Illustration of using DNNs for both the source acoustic model and the phone mapping (*Setup C*).

layer is to enforce the network to pass useful information that discriminates different classes of source language through the bottleneck layer. The BN-DNN well-trained by the source language training data is used to generate cross-lingual bottleneck feature for different target language acoustic models such as HMM/GMM or HMM/DNN. The performance of this setup will be compared with our proposed phone mapping method (*Setup A*) in Section 4.2.2.

4.2.1.3 Setup C: using DNNs to implement the phone mapping function

Besides using DNN-based acoustic model for the source language, in this setup, DNNs are also used to replace MLPs as the cross-lingual phone mapping which is called “deep phone mapping”. In the *Setup A*, although DNNs are used as the source language acoustic model, 3-layer MLPs are still used to realize the phone mapping function since it is hypothesized that cross-lingual posteriors are higher level features than raw features such as MFCCs and hence we could use a simple mapping. However, experimental results in Chapter 3 (Section 3.3.4.5) showed that using 3-layer MLP to realize phone mapping achieves a significant improvement over simpler phone mappings such as linear combination or 2-layer neural networks. In addition, the study in Section 4.1.4 has

shown that using deep structures can help to consistently improve performance of the DNN acoustic model for monolingual tasks even under limited training data conditions. In this setup, we will answer an important question, how much target language training data are required for effective training of a deep phone mapping. The combination of the DNN-based source acoustic model in *Setup A* and the DNN-based phone mapping is shown in Fig. 4.8. From the figure, it is shown that there are two DNNs, the output of the source language DNN acoustic model is directly fed into the input of the DNN phone mapping.

4.2.2 Experiments

In this section, experiments for the above three setups are conducted. The section is organized as follows. Subsection 4.2.2.1 introduces databases, neural network architectures and other experimental setup. Subsection 4.2.2.2 presents the experimental results using deep source acoustic model in phone mapping (*Setup A*). Subsection 4.2.2.3 conducts cross-lingual acoustic models using bottleneck feature generated by the source language bottleneck network (*Setup B*) for comparison purpose. Subsection 4.2.2.4 presents the experimental results when deep structures are applied for both the source language acoustic model and the phone mapping (*Setup C*).

4.2.2.1 Experimental setup

Task and Databases:

All experiments in this chapter use the same databases as in Chapter 3. Specifically, Malay is used as the source language and English is the target language. The experiments are to measure the word error rate (WER) on the target language using three setups. In addition to three limited training data sizes i.e. 7, 15, 55 minutes of the target language as in Chapter 3, we also generate a 220 minute data set to examine whether we can benefit from cross-lingual phone mapping when even more training data are available.

Neural networks:

Two 5-layer DNNs i.e. POS-DNN in *Setup A* and BN-DNN in *Setup B* are used as the source language acoustic models to generate cross-lingual posterior and bottleneck features, respectively. The input of each DNN is formed by concatenating 9 frames of

39-dimensional MFCCs. Each hidden layer of both the DNNs consists of 2000 hidden units except the middle hidden layer of the BN-DNN i.e. the bottleneck layer has 39 units. Chapter 3 has shown that using context-dependent triphone in the source language acoustic models results in better phone mapping performance. Hence, the source POS-DNN in this case is also a context-dependent DNN i.e. it consists of 1592 outputs to represent 1592 tied-states in the source acoustic model. While using source language context-dependent POS-DNN for phone mapping (*Setup A,C*) achieve a better performance, our experiments indicated that using context-dependent BN-DNN does not achieve any improvement over context-independent BN-DNN for cross-lingual speech recognition (*Setup B*). Hence, for the source language BN-DNN, 102 outputs are used to represent 102 monophone states in the Malay source acoustic model.

Both the POS-DNN and BN-DNN are first initialized by RBM pre-training [42]. For the first RBM (i.e. Gaussian-Bernoulli), a learning rate of 0.005 is used with 10 pre-training epochs. The rest RBMs (i.e. Bernoulli-Bernoulli), a learning rate of 0.05 is used with 5 pre-training epochs. After that the DNNs are fine tuned by the back-propagation algorithm. The “newbob”² procedure is used with the initial learning rate of 0.008.

For the target language neural networks, since the target training data are limited, small-scaled neural networks are used with 500 units for each hidden layer. 3-layer architecture, i.e. with 1 hidden layer, is used for the case of shallow networks and 5-layer architecture, i.e. with 3 hidden layers, is used for the case of deep networks.

4.2.2.2 Setup A: using DNNs as the source language acoustic model to generate cross-lingual posterior feature for phone mapping

Fig. 4.9 shows the WER given by three models for four different amounts of target training data where the two first models are used as the baselines for comparison:

- (i) Monolingual HMM/DNN model with MFCC feature (presented in Section 4.1.4.2).
- (ii) Cross-lingual phone mapping using shallow source language MLP acoustic model (proposed in Chapter 3, Section 3.3.4.6).

²<http://www1.icsi.berkeley.edu/Speech/faq/nn-train.html>

- (iii) Cross-lingual phone mapping using deep source language DNN acoustic model (*Setup A*).

It is observed that using DNN as the source acoustic model can improve significantly the performance of phone mapping over using shallow models. It shows that DNN can model the source language better than shallow models, and hence posteriors generated from DNN are better and then result in better performance of phone mapping. It is also seen that both the two cross-lingual phone mappings significantly outperform the monolingual model for limited training data conditions. This gain however reduces when more target training data are available. For the case of 220 minutes, cross-lingual phone mapping with shallow source acoustic model even achieves a worse WER than the monolingual HMM/DNN model while cross-lingual phone mapping using deep source acoustic model still obtains slightly better performance than the baseline monolingual HMM/DNN model (7.5% versus 7.8% WER). It shows that when more target language training data are available, if the source language model is not strong enough (i.e. shallow), it can result in the poor performance of phone mapping.

4.2.2.3 Setup B: using DNNs to extract cross-lingual bottleneck feature for target language acoustic models

As shown in Fig. 4.7, cross-lingual bottleneck feature generated by the source language bottleneck network can be used for various target language acoustic models such as HMM/GMM or HMM/DNN. Unlike the cross-lingual tandem approach presented in Section 3.3.4.3.c in Chapter 3, using bottleneck feature with the target HMM/GMM model, we do not need to perform dimensionality reduction such as PCA since bottleneck feature is low dimensional features which can fit well to the HMM/GMM model.

Fig. 4.10 shows the WER of the two cross-lingual models using bottleneck feature generated by the source language bottleneck network (BN-DNN). The first and second columns represent the WER given by the two cross-lingual systems which use target language HMM/GMM and HMM/DNN models, respectively. It is observed that using target HMM/DNN acoustic model results in a significantly better performance over using target HMM/GMM model especially under limited training data conditions. However, both of the cross-lingual models using source bottleneck feature are outperformed by the

Figure 4.9: WER (%) of the cross-lingual phone mapping using deep source acoustic model versus the phone mapping using shallow source acoustic model and the monolingual model.

Figure 4.10: WER (%) of the two cross-lingual models using bottleneck feature generated by the deep source bottleneck network (*Setup B*). The last column shows the WER given by the phone mapping model in *Setup A* for comparison.

proposed phone mapping (*Setup A*) in the last column. The reason can be explained that placing a bottleneck layer before the output layer can reduce performance of the neural network [101]. Our recent study [30] also confirmed that using bottleneck neural network (BN-DNN) can reduce the frame accuracy at the output layer over the conventional DNN (POS-DNN) for both the training and development set. This may cause a performance degradation in cross-lingual models.

4.2.2.4 Setup C: using DNNs to implement the phone mapping function

The experiment in this section is conducted to answer two questions: (i) can improvement be achieved using a deep phone mapping? (ii) would RBM pre-training initialization help to improve over random weight initialization for the deep phone mapping?. Hence, three different configurations for phone mapping are conducted:

- Shallow phone mapping: MLP with random weight initialization (*Setup A*).
- Deep phone mapping (rand): DNN with random weight initialization (*Setup C*).
- Deep phone mapping (RBM): DNN with RBM pre-training initialization (*Setup C*).

The comparison result of using the above three different configurations is showed in Fig. 4.11. We can see that using deep structures for phone mapping is only useful if sufficient training data are available i.e. 55 and 220 minutes. There is no different between random weight initialization and RBM pre-training initialization in this case. We also find that if keeping the amount of training data for back-propagation i.e. supervised training fixed, using more training data for unsupervised RBM pre-training, no further improvement is observed. This observation is contrary to the result in Section 4.1.4.2 where DNNs were superior to MLPs for a monolingual task even under very limited training data conditions. It can be explained that the mapping from source language posterior feature to the target phone states is easier to implement than the mapping from low-level feature MFCC to the target phone states as in the monolingual model. Hence, under limited training data conditions, it does not require deep structures or RBM pre-training for cross-lingual phone mapping.

Figure 4.11: Comparison of the cross-lingual phone mapping using shallow and deep structures for phone mapping.

4.3 Conclusion

This chapter improved the phone mapping framework proposed in Chapter 3 by using deep neural networks (DNNs). Two approaches have been investigated. In the first approach, DNNs are used as the source language acoustic model. The experimental results showed that using DNN source language acoustic models produced significantly better results than using shallow source language acoustic models in the proposed cross-lingual phone mapping framework. This suggests that cross-lingual posteriors generated from deep models are of higher quality than those generated from shallow models. In addition, we also found that using cross-lingual posteriors as in the phone mapping method is a better choice than using cross-lingual bottleneck feature for cross-lingual speech recognition. In the second approach, DNNs are used to realize the phone mapping function in place of shallow MLPs. Unlike the first approach, using DNNs for phone mapping is only useful when sufficient target training data are available. In this approach, no improvement is found by using RBM pre-training initialization over random weight initialization even more training data are used for unsupervised pre-training of the phone mapping.

In conclusion, in the phone mapping framework, we need a strong source language acoustic model to generate better cross-lingual posterior features for the target language speech data. However, the mapping from source posteriors to the target posteriors is a simple task, it can be implemented using a shallow network under limited training data conditions.

Chapter 5

Exemplar-based Acoustic Models for Limited Training Data Conditions

This chapter focuses on building a robust acoustic model using exemplar-based modeling technique under limited training data conditions. Unlike conventional acoustic models presented in the previous chapters, such as GMM or DNN, exemplar-based model is non-parametric and uses the training samples directly to form the model parameters. Hence, the approach does not assume a parametric form for the density or discriminant functions and is an attractive method when the functional form or the distribution of decision boundary is unknown or difficult to estimate under limited training data conditions [70].

In this chapter, a specific exemplar-based model, called kernel density estimation is used to generate the likelihood of target language triphone states. However, we found that simply using MFCC feature for the kernel density model results in a poor speech recognition performance [31]. This is because the distance between test MFCC feature vectors and exemplar vectors is simply the Euclidean distance, which is not robust to the variations of low-level features such as MFCC. To address this, three approaches are proposed. Firstly, the higher level cross-lingual bottleneck feature is used as the input for the kernel density model. Secondly, a novel Mahalanobis distance based metric, optimized by minimizing the classification error rate on the training data, is proposed. Thirdly, a discriminative score tuning network to fine tune the likelihood scores by also minimizing training classification error is suggested. Experimental results on the Wall Street Journal (WSJ) corpus show that the proposed kernel density model achieves improved performance and performs even better than the DNN acoustic model under limited training data conditions.

The work in this chapter is published in INTERSPEECH 2014 [31].

5.1 Introduction

This section provides a brief introduction of the exemplar-based model and its application for speech recognition. In addition, the three proposed approaches to improve exemplar-based acoustic model for limited training data conditions are also discussed.

Unlike the parametric methods for acoustic modeling such as GMM or DNN, exemplar-based method is a non-parametric model that uses the training samples directly to form the model e.g. the k-nearest neighbors (k-NN) method [70] for classification and kernel density (or Parzen window) [70] for density estimation are well known examples. Although exemplar-based methods such as k-NN are very simple and easy to implement, they are often among the best performing techniques in many machine learning tasks [73]. Exemplar-based methods have been popular in many tasks such as face recognition [71], object recognition [72] and audio classification [102].

Recently, several studies have applied exemplar-based methods for acoustic modeling [74–77]. In [76], the authors proposed a method to learn class label embeddings to model the similarity between labels within a nearest neighbor framework. They applied these estimates to acoustic modeling for speech recognition. Experimental results showed that significant improvements in terms of word error rate (WER) are achieved on a lecture recognition task over a state-of-the-art baseline GMM model. In [74], the authors reported that using kernel density estimation can achieve promising results under limited training data conditions. The k-NN method was also used in [76], a consistent improvement is achieved when applying smoothing across frames. Such results motivated us to apply exemplar-based methods to resource-limited LVCSR task, where the amount of training data is very little.

This chapter applies the exemplar-based approach at the frame-level for acoustic modeling of resource-limited speech recognition. Specifically, we apply kernel density estimation [74, 80] to replace GMM of the HMM/GMM model or DNN of the HMM/DNN model to estimate the state likelihood probability. This approach is selected since it fits in the current LVCSR system with modest changes. In our experiments, we however

found that simply using the kernel density model with conventional MFCC feature yields worse results than GMM [31]. We suggest that the simple Euclidean distance used in kernel density estimation is not robust against MFCC feature variations. To address this, three approaches are proposed: using cross-lingual bottleneck feature in place of MFCC, applying distance metric learning in place of the Euclidean distance and using discriminative score tuning.

- (i) In the first approach, cross-lingual bottleneck feature generated by a bottleneck neural network trained by a source language [16, 17, 30, 61] is used as the acoustic feature for the kernel density model of the target language. Cross-lingual bottleneck feature has been used effectively for HMM/GMM models [16], hybrid HMM/MLP [30] as well as in our recent kernel density model [31]. Our results show that when the target language training data are limited, the kernel density model using cross-lingual bottleneck feature achieves a significant improvement over the conventional features such as MFCC.
- (ii) In the second approach, distance metric learning is applied to learn the distance metric between a test vector and a training exemplar to improve speech recognition performance. In other words, a transformation is applied to transform input feature to a new space where the Euclidean distance can be performed well [103]. Specifically, a Mahalanobis based distance is learnt in an iterative manner to improve the frame classification accuracy on the training set by maximizing the mutual information criterion (MMI) [104].
- (iii) In the third approach, we address a limitation of the kernel density method that it tries to estimate the distribution of the speech classes rather than optimal decision boundary between classes e.g. HMM states. Hence, its performance is not optimal in terms of speech recognition accuracy. To solve this, we introduce a discriminative score tuning module on the top of the kernel density estimation and the speech recognition performance is consistently improved.

The rest of this chapter is organized as follows: Section 5.2 presents the kernel density acoustic model framework using the Euclidean distance. Section 5.3 describes the proposed distance metric learning method. Section 5.4 shows the proposed discriminative

score tuning module. Section 5.5 presents the experimental settings and results. Finally, we conclude in Section 5.6.

5.2 Kernel Density Model for Acoustic Modeling

In this study, instead of using a GMM to model the feature distribution for a triphone tied-state as in the conventional HMM/GMM acoustic model, the kernel density model similar to the one used in [31, 74] is applied. Specifically, the likelihood of a feature vector, $\mathbf{o}_t$ for a speech class, s_j , i.e. a tied-state is estimated as follows:

$$p(\mathbf{o}_t | s_j) = \frac{1}{ZN_j} \sum_{i=1}^{N_j} \exp\left(-\frac{\|\mathbf{o}_t - \mathbf{e}_{ij}\|^2}{\sigma}\right) \quad (5.1)$$

where $\mathbf{e}_{ij}$ is the i^{th} exemplar of class j , $\|\mathbf{o}_t - \mathbf{e}_{ij}\|^2$ is the Euclidean distance between $\mathbf{o}_t$ and $\mathbf{e}_{ij}$, σ is a scale variable, N_j is the number of exemplars in class j , and Z is a normalization term to make Eq. (5.1) be a valid distribution.

From Eq. (5.1), the likelihood function is mathematically similar to a GMM with a shared scalar variance for all dimensions. Effectively, Eq. (5.1) puts a Gaussian-shaped function at each training exemplar and sums all these Gaussians with a normalization factor to the likelihood function. The likelihood function in Eq. (5.1) is a non-parametric way to estimate the distribution of the features, i.e. we do not assume any structure for the density and the model size grows with the training data. However, the limitation of the kernel density estimation approach as compared to the GMM approach is its high computational cost in the decoding process as it requires computation of the distance between every testing frame with all frames in the training set. As this thesis focuses on building the kernel density model with limited training data, the computational cost will not be a big problem. However, when training data increase, pruning techniques should be considered to accelerate the decoding process.

As shown in Fig 5.1, there are four main steps to build an LVCSR system with the kernel density acoustic model [31]:

Step 1 Build a triphone-based HMM/GMM acoustic model.

Figure 5.1: Kernel density estimation for acoustic modeling of an LVCSR system.

Step 2 Generate state label for each frame (exemplar) of training data using forced alignment. The training exemplars are then grouped based on their state label.

Step 3 Use kernel density model to estimate HMM state emission probability $p(o_t | s_j)$ as in Eq. (5.1)¹.

Step 4 Plug the state emission probability into a standard decoder such as Viterbi decoder for decoding.

In our preliminary experiments, we found that using MFCC as the input feature on the kernel density model results in worse performance than the HMM/GMM baseline [31]. This could be due to the fact that the kernel density model of Eq. (5.1) uses the Euclidean distance with global scale variable σ to measure distance between test features and exemplars. This distance metric may not handle well the MFCC feature variations. To address this, we use cross-lingual bottleneck feature which can be considered as a

¹In fact, scaled likelihood is used since the normalization term, Z in Eq. (5.1) is the same for all classes and never needs to be computed [31].

high-level discriminative feature for the kernel density model. Our experiments achieved promising performance under limited training data conditions [31]. This issue will be investigated in details in Section 5.5.2.

Another issue of the kernel density acoustic model in Fig. 5.1 is that it uses the Euclidean distance which treats all feature dimensions equally and ignores the correlation between feature dimensions. Hence our second extension discussed below is to use a more meaningful distance metric in the kernel density model to address this problem.

5.3 Distance Metric Learning

Several distance metric learning approaches have been proposed for different applications. For example, the large margin nearest neighbors (LMNN) algorithm [105] is a supervised approach to learning Mahalanobis distance metric. LMNN seeks a linear feature transformation such that, in the transformed space, the k nearest exemplars from the correct class and exemplars from other classes become separated by a larger margin. Another metric learning technique is the locality preserving projections (LPP) algorithm [106]. LLP learns a linear transformation $\mathbf{Q} : \mathcal{R}^d \mapsto \mathcal{R}^p$ where ($p \leq d$) that aims to preserve the neighborhood structure of the data. In this study, we apply metric learning to kernel density based acoustic model and learn a Mahalanobis based distance metric that is optimized for speech recognition. A Mahalanobis based distance, which is equivalent to a full rank linear feature transformation, is learnt to optimize the frame classification accuracy on the training.

A Mahalanobis based distance is defined as [103]:

$$d(\mathbf{o}_t, \mathbf{e}_{ij}) = (\mathbf{o}_t - \mathbf{e}_{ij})^T \mathbf{M} (\mathbf{o}_t - \mathbf{e}_{ij}) \quad (5.2)$$

where $d(\mathbf{o}_t, \mathbf{e}_{ij})$ is the Mahalanobis based distance between input feature vector $\mathbf{o}_t$ and exemplar $\mathbf{e}_{ij}$, $\mathbf{M}$ is a matrix to be learnt. Since $\mathbf{M}$ is a symmetric positive semi-definite matrix, it can be factored as: $\mathbf{M} = \mathbf{Q}^T \mathbf{Q}$. Hence Eq. (5.2) can be rewritten as

$$d(\mathbf{o}_t, \mathbf{e}_{ij}) = (\mathbf{Q}\mathbf{o}_t - \mathbf{Q}\mathbf{e}_{ij})^T (\mathbf{Q}\mathbf{o}_t - \mathbf{Q}\mathbf{e}_{ij}). \quad (5.3)$$

This implies the Mahalanobis based distance can be interpreted as the Euclidean distance in a transformed space, $\mathbf{o}_t \rightarrow \mathbf{Q}\mathbf{o}_t$ [103]. The purpose of this work is to learn transformation $\mathbf{Q}$ to transform the input space to a new space where the kernel density model can perform better.

The proposed method learns a Mahalanobis based distance in Eq. (5.3) to optimize the frame accuracy on the training set by maximizing posterior probability $p(s_C|\mathbf{o}_t)$ of the correct HMM state s_C for each input feature $\mathbf{o}_t$. In this work, for each $\mathbf{o}_t$, the cost function f , which is a function of transformation $\mathbf{Q}$, is defined as the MMI (Maximum Mutual Information) criterion as follows:

$$f(\mathbf{Q}) = \log(p(s_C|\mathbf{o}_t)) = \log\left(\frac{p(\mathbf{o}_t|s_C)p(s_C)}{\sum_{j=1}^J p(\mathbf{o}_t|s_j)p(s_j)}\right) \quad (5.4)$$

where J is the number of states, s_C is the correct state label for input vector $\mathbf{o}_t$, $p(s_j)$ is the state prior estimated from training data, $p(\mathbf{o}_t|s_j)$ is the likelihood probability estimated by the kernel density model as in Eq. (5.5) for input feature $\mathbf{o}_t$ and state s_j ,

$$p(\mathbf{o}_t|s_j) = \frac{1}{ZN_j} \sum_{i=1}^{N_j} \exp\left(-(\mathbf{Q}\mathbf{o}_t - \mathbf{Q}\mathbf{e}_{ij})^T(\mathbf{Q}\mathbf{o}_t - \mathbf{Q}\mathbf{e}_{ij})\right). \quad (5.5)$$

Eq. (5.5) is derived from Eq. (5.1) by using the Mahalanobis based distance defined in Eq. (5.3) and the scaling factor σ is set to 1.

The goal is to find $\mathbf{Q}$ to maximize $f(\mathbf{Q})$. In this study, the gradient descent algorithm is applied to update $\mathbf{Q}$ iteratively. The proposed distance metric learning procedure is described as follows:

Step 1 Initialize transformation $\mathbf{Q} \in \mathcal{R}^{D \times D}$ as an identity matrix where D is the dimension of the input feature, $\mathbf{o}_t$.

Step 2 Compute the derivative of f with respect to $\mathbf{Q}$, $\frac{\partial f}{\partial \mathbf{Q}}$. The detailed derivation is presented at Appendix A.1.

Step 3 Update $\mathbf{Q}$ using the gradient descent method: $\mathbf{Q}^{new} \leftarrow \mathbf{Q}^{old} + \alpha \frac{\partial f}{\partial \mathbf{Q}}$, where α is the learning rate.

Figure 5.2: The proposed discriminative score tuning.

Step 4 Estimate the likelihood scores of all samples in the development set using Eq. (5.5). Convert these scores into state posteriors to compute the frame accuracy in the next step.

Step 5 The recognized state label for each sample (frame) in the development set is determined by picking the state label of the highest posterior score in that frame. Compute the frame accuracy in the development set by comparing the recognized state label and the ground truth, i.e. state label provided by forced alignment in the development set. If the frame accuracy still significantly increases go back Step 2, otherwise, we quit the learning procedure.

5.4 Discriminative Score Tuning

The class likelihood function in Eq. (5.5) has two limitations when it is used for speech recognition. Although the likelihood function asymptotically approaches the true density when infinite training data are available, in the case of very few training data, it may not lead to good recognition performance as the density estimate is not robust. Secondly, the dynamic range of the log likelihood generated by Eq. (5.5) may be very different from the conventional GMM system, making it necessary to carefully re-tune the language model scale and beam width. Such tuning process is tedious and may not produce the best results. In this section, we propose a discriminative score tuning step to address these two limitations.

A neural network can be used to tune the likelihood scores with the criterion of frame classification accuracy. The proposed discriminative tuning module is illustrated in Fig. 5.2. The figure shows that state likelihood score $p(\mathbf{o}_t|s_j)$ generated by the kernel density model is first converted to state posterior score $p(s_j|\mathbf{o}_t)$ using the Bayes rule:

$$p(s_j|\mathbf{o}_t) = \frac{p(\mathbf{o}_t|s_j)p(s_j)}{\sum_{j'=1}^J p(\mathbf{o}_t|s_{j'})p(s_{j'})}. \quad (5.6)$$

where state prior $p(s_j)$ is estimated from training data, J is the number of HMM tied-states. Posterior score $p(s_j|\mathbf{o}_t)$ is then fine tuned by a neural network. The number of inputs and outputs of the neural network is the same, and is equal to the number of HMM tied-states, J . The goal of the neural network is to estimate a new posterior $p'(s_j|\mathbf{o}_t)$ such that it maximizes the frame classification accuracy.

Since the standard decoder uses state likelihood scores in the decoding process, state posteriors after tuning must be converted back to likelihood scores. In practice, scaled likelihoods are used as in Eq. (5.7) since the scaling factor $p(\mathbf{o}_t)$ is constant for all states and does not affect the classification decision.

$$p'(\mathbf{o}_t|s_j) = \frac{p'(s_j|\mathbf{o}_t)p(\mathbf{o}_t)}{p(s_j)} \propto \frac{p'(s_j|\mathbf{o}_t)}{p(s_j)}. \quad (5.7)$$

5.5 Experiments

5.5.1 Experimental procedures

Similar to Chapter 3 and Chapter 4, we evaluate the performance of the proposed method on the WSJ0 task. Four different training data sizes are used in this study. For each training size, an HMM/GMM system is first built with a different number of tied-states. This number is determined empirically. Specifically, we use 243, 243, 501 and 1005 tri-phone tied-states for the cases of 7, 16, 55 and 220 minutes of training data, respectively. For each data size, we randomly extract around 10% amount of data from the training set to build the development set. The rest is used to train the models. The baseline HMM/GMM model provides both the state-tying decision tree and frame level state label for the building of the hybrid HMM/DNN and the kernel density (HMM/KD) models. Since the training data sizes are small, a small-scaled DNN architecture with 5-layer structure and 500 neurons in each hidden layer serving as a competitive baseline model is built. The DNNs are initialized using RBM pre-training [42]. The test data are 166 clean utterances, or about 20 minutes of speech.

Similar to Chapter 4, two types of feature are investigated in this chapter. The first feature is 39 dimensional MFCC, including 13 static features and its time derivatives. The second feature is the chapter's first approach to use cross-lingual bottleneck feature

for the kernel density model. Bottleneck feature is generated by a bottleneck DNN which is well-trained using Malay speech corpus. More detailed setup can be found in Section 4.2.1.2 of Chapter 4 (*Setup B*).

In this work, the focus is on acoustic model training with limited training data hence we assumed that the language model and pronunciation dictionary are available. The standard WSJ bigram LM and the 5k vocabulary are used in decoding. In the hybrid model (HMM/DNN) and the kernel density model (HMM/KD), for each HMM state, the probability of jumping to the next state is simply set to 0.5.

5.5.2 Using MFCC and cross-lingual bottleneck feature as the acoustic feature

This section reports the performance of the first approach of using cross-lingual feature for the kernel density model. The results in word error rate (WER) obtained by various models using MFCC and cross-lingual bottleneck features with four different amounts of training data are presented in Table 5.1.

The first and second rows show the result of the two baseline models i.e. conventional HMM/GMM and hybrid HMM/DNN using MFCC features (reported in Section 4.1.4.2 of Chapter 4). As expected, the WER gets worse when less training data are used. The HMM/DNN model outperforms the HMM/GMM model significantly for all training data sizes.

The third row of Table 5.1 shows the results obtained by using MFCC feature with the plain HMM/KD model presented in Section 5.2, i.e. without distance metric learning and score tuning. In this experiment, scaling factor σ in Eq. (5.1) is set to 1. Unfortunately, the kernel density model produces worse results than the HMM/GMM baseline. As discussed in Section 5.2, the reason is that the Euclidean distance is not robust to feature variation of MFCC.

Next, let's examine the results obtained by using the cross-lingual bottleneck feature. In Table 5.1, row 7 and 8 show the WER of the two baseline HMM/GMM and HMM/DNN models using cross-lingual bottleneck feature (reported in Section 4.2.2.3 of Chapter 4). It is observed that using cross-lingual bottleneck feature significantly improves both the HMM/GMM and HMM/DNN models especially for the case of very limited target language training data. These results show the benefit of using cross-lingual

Table 5.1: WER (%) obtained by various models at four different training data sizes. Row 1-6 are results obtained by using MFCC feature. Row 7-11 show results obtained by using cross-lingual bottleneck feature. KD stands for kernel density used for acoustic modeling.

No	Acoustic model	Training data (minutes)			
		7	16	55	220
Monolingual model (MFCC feature)					
1	HMM/GMM baseline model	30.9	23.1	14.0	9.1
2	HMM/DNN baseline model	24.1	17.9	11.3	7.8
3	Plain HMM/KD	33.7	26.2	15.5	11.5
4	HMM/KD+LDA	33.3	25.6	15.1	11.0
5	HMM/KD+distance metric learning (DML)(<i>approach 2</i>)	25.6	19.5	11.5	8.3
6	HMM/KD+DML+score tuning (<i>approach 2+3</i>)	23.3	17.1	9.7	7.0
Cross-lingual model (cross-lingual bottleneck feature) (<i>approach 1</i>)					
7	HMM/GMM baseline model	24.6	18.5	11.3	10.1
8	HMM/DNN baseline model	17.5	15.3	10.3	8.5
9	Plain HMM/KD (<i>approach 1</i>)	18.8	15.8	10.6	8.2
10	HMM/KD+distance metric learning (DML)(<i>approach 1+2</i>)	18.4	15.6	10.3	8.0
11	HMM/KD+DML+score tuning (<i>approach 1+2+3</i>)	15.7	13.1	9.8	7.6
Phone mapping using deep source acoustic model (Setup A - Section 4.2.2.2)					
12	Phone mapping using deep source acoustic model	16.3	13.9	8.9	7.5

features generated by the well-resourced language models. Note that at 220 minutes, the HMM/GMM and HMM/DNN with cross-lingual bottleneck feature actually perform worse than those models with MFCC feature. This could be due to that the gain of cross-lingual bottleneck feature is offset by the loss due to the mismatch between the two corpora i.e the source language and the target language. Now, we focus on row 9 of Table 5.1 when the cross-lingual bottleneck feature is used for the HMM/KD model. A large improvement is observed over the result in row 3 where MFCC feature is used. Moreover, in the case of using the cross-lingual bottleneck feature, the HMM/KD model outperforms the HMM/GMM model in row 7 significantly and even approaches to the HMM/DNN model in row 8. This shows that the HMM/KD model can be well employed when good input features are used.

5.5.3 Distance metric learning for kernel density model

5.5.3.1 Experimental results

The experiments in Section 5.5.2 indicated that the plain kernel density model with the Euclidean distance is outperformed by the conventional HMM/GMM model when MFCC is used as the input feature. However, using cross-lingual bottleneck feature, the kernel density model achieves a significantly better performance than the HMM/GMM model. To improve the kernel density model with MFCC feature, the Euclidean distance is replaced by the Mahalanobis based distance metric as in Eq. (5.3). Transformation matrix $\mathbf{Q}$ is trained following the procedure in Section 5.3. We use the mini-batch mode update strategy with the mini-batch size of 50, i.e. $\mathbf{Q}$ is updated after every 50 frames using accumulative gradient. The learning rate, α is set to 0.0002.

The result of the kernel density model with the distance metric learning for MFCC feature is listed in row 5 of Table 5.1. A large improvement is achieved over the plain kernel density model in row 3. This improvement remains stable over different training data sizes i.e. from 24% to 28% relative. The result in row 5 is also significantly better than the HMM/GMM model in row 1 and approaches the performance of the HMM/DNN model in row 2.

For comparison, we also apply Linear Discriminant Analysis (LDA) which uses the label of training data to linearly separate classes. In this experiment, MFCC is applied LDA to keep all 39 dimensions before used as the input for the conventional kernel density model. As shown in row 4 of Table 5.1, using LDA just results in a small improvement in WER over the plain kernel density model with MFCC in row 3. The study in [107] indicated that LDA suffers from a small sample size problem when dealing with high-dimensional data. Our experimental results show that LDA can give 4.3% relative improvement for the case of 220 minutes, this improvement drops to 1.2% when 7 minutes of training data are used. In another experiment, we use the standard Mahalanobis distance as in Eq. (5.2) where $\mathbf{M} = \mathbf{S}^{-1}$ with $\mathbf{S}$ is the covariance matrix of input feature $\mathbf{o}_t$. However, the result is even not as good as using the Euclidean distance and hence is not reported here.

We also apply the proposed distance metric learning for cross-lingual bottleneck feature, the result is shown in row 10 of Table 5.1. Unlike to the case of MFCC feature,

Figure 5.3: Illustration of the linear feature transformation matrices. BN stands for cross-lingual bottleneck features, DML stands for distance metric learning. The MFCC feature vectors are ordered as $[c1, \dots, c12, c0]$, and their delta and acceleration versions.

applying distance metric learning for bottleneck feature achieves only a small improvement over the plain kernel density model in row 9. It indicates that distance metric learning is more important when low level features such as MFCC are used as the input of the kernel density model.

5.5.3.2 Distance metric learning illustration

To give an insight to why the proposed distance metric learning approach can improve performance of the kernel density acoustic model, we compare the feature transformations learnt by LDA, and distance metric learning on MFCC and cross-lingual bottleneck features. The MFCC feature transformation learnt by LDA is shown in Fig. 5.3(a). For comparison, the MFCC feature transformation learnt by the proposed distance metric learning is shown in Fig. 5.3(b). It can be observed that the transformation learnt by distance metric learning has an obvious diagonal structure. From the values of the diagonal elements, the weights of MFCC features $c0$ - $c12$ are almost monotonically decreasing, meaning that in the kernel density model, lower order MFCC features are more important than higher order MFCC features for frame classification. It is also observed that there are two off diagonal bars in the transformation, which model the correlation between static and corresponding acceleration features. On the other hand, there is no clear structure in the LDA-derived transformation matrix.

The learnt transformation matrix for cross-lingual bottleneck feature is shown in Fig. 5.3(c). It is observed that the transformation matrix for the bottleneck feature is closer

to the identity matrix than the MFCC transformation matrix, although both are learnt by the distance metric learning method. The diagonal values of the transformation for bottleneck feature are similar to each other, meaning that all dimensions of bottleneck feature contribute similarly to the kernel density model and hence speech recognition. In addition, there is no clear off-diagonal structure, this may indicate that there is no obvious correlation between dimensions of bottleneck feature that can be modeled for better frame classification performance. The above observations are reasonable considering that cross-lingual bottleneck feature is extracted by a deep neural network that is trained to discriminate sound classes. Hence there is less gain to apply the distance metric learning method on bottleneck feature than on MFCC feature.

5.5.3.3 Effects of Q initialization

In the previous distance metric learning experiments, the transformation matrix Q is initialized as an identity matrix. Now we examine how the initialization procedure can affect the performance. In this experiment, Q is initialized randomly. Specifically, each value of Q is set to a small random value. The result of the kernel density model using MFCC feature with two initialization schemes for Q is listed in Table 5.2. We can see that there is almost no difference for the two initialization schemes in term of speech recognition performance. However, the difference comes from the number of required iterations to train Q . Fig. 5.4 shows the frame error rate (FER) of tied-states for the development set given by the kernel density model over the training process of Q for the case of 16 minutes training data. We can see that finally both of the two initialization schemes provide a similar FER. However, by using an identity matrix as the initialization for Q , we can obtain Q just after few iterations. Note that the training process of distance metric learning is very time consuming hence all next experiments will use an identity matrix for Q initialization.

5.5.4 Discriminative score tuning

The likelihood scores generated by the kernel density model are further refined by the score tuning module as described in Section 5.4. Specifically, a 2-layer neural network is placed on the top of the kernel density model to fine tune the likelihood scores. This

Table 5.2: WER (%) obtained by the kernel density model using MFCC feature with two initialization schemes for transformation Q .

No	Initialization	Training data (minutes)			
		7	16	55	220
1	Identity matrix (row 5 of Table 5.1)	25.6	19.5	11.5	8.3
2	Random-weight-matrix	25.5	19.6	11.7	8.2

Figure 5.4: Frame error rate (%) obtained by the kernel density model with two initialization schemes for transformation Q .

neural network is trained with the cross-entropy criterion to minimize the training frame classification error. While the distance metric learning makes more discriminative feature for the kernel density model, score tuning aims to fine tune the likelihood scores in a discriminative manner. The results of the kernel density model with distance learning and score tuning are shown in row 6 of Table 5.1 for the case of MFCC input and row 11 for the case of cross-lingual bottleneck feature input. It can be seen that using the discriminative score tuning can significantly improve the performance of the kernel density model for both the cases when MFCC and bottleneck features are used. The kernel density model with distance metric learning and score tuning provides the best performance over the GMM and even the DNN models for all four training data sizes. The experimental results also show that when training data are more than 55 minutes, the proposed kernel density model with MFCC feature achieves better results than with

Figure 5.5: The weight matrix of the 2-layer neural network score tuning.

cross-lingual bottleneck feature.

Row 12 of Table 5.1 shows the result of the phone mapping approach using deep models in Chapter 4 for comparison purpose. It can be seen that under very limited training data conditions i.e. 7 and 16 minutes, the cross-lingual exemplar-based method in row 11 consistently outperforms the phone mapping method. However, when more target training data are available the exemplar-based is outperformed by the phone mapping approach. Also note that this is not a really fair comparison since the phone mapping uses cross-lingual posterior feature while the exemplar-based model uses cross-lingual bottleneck feature. In the future work, cross-lingual posterior feature will be considered as the input feature to the exemplar-based system.

Fig. 5.5 illustrates the input to output weight matrix of the 2-layer-neural network score tuning in the case of using 16 minutes of training data. In this case the acoustic model contains 243 tied-states. We can see diagonal elements have much larger values than the off-diagonal elements. This shows that the neural network just slightly tunes the posterior scores and the output scores have very high correlation with the input posteriors. This is reasonable as the input and output are the posteriors of the same classes.

Figure 5.6: Performance in WER(%) of the proposed kernel density model using different score tuning neural network architectures.

Now, we examine whether we can benefit from using more complicated neural network architectures for score tuning. To do this, a 3-layer neural network (i.e. 1 hidden layer) and a 5-layer neural network (i.e. 3 hidden layers) are used for score tuning. Each hidden layer contains 500 neurons. Fig. 5.6 illustrates the WER obtained by the proposed kernel density model with and without score tuning for two different input feature i.e. MFCC and cross-lingual bottleneck features. The first column is the WER provided by the kernel density model without score tuning (shown in row 5 and row 10 of Table 5.1). The second column shows the WER of the kernel density model with 2-layer neural network score tuning (shown in row 6 and row 11 of Table 5.1). Column 3 and 4 present the WER of the kernel density model using 3- and 5-layer neural network architectures for score tuning, respectively. It can be seen that different neural network architectures for score tuning provide a similar performance for the kernel density model.

5.6 Conclusion

This chapter studied the development of an acoustic model for resource limited LVCSR using kernel density model. Three approaches to improve performance have been proposed. First, the cross-lingual bottleneck was used to replace MFCC as the input for the kernel density model. Second, a Mahalanobis-based distance metric to replace the Euclidean distance was explored. Third, a score tuning method was applied to fine tune the likelihood scores using a discriminative criterion. With these three approaches, the performance of the kernel density model is improved. The experimental results on the WSJ corpus showed that the proposed model produces consistently better results than both the HMM/GMM and HMM/DNN models, up to 220 minutes of training data.

One issue of using the kernel density model is its high computational cost for decoding especially when training data increase. To reduce the decoding time, pruning techniques will be investigated in the future works.

Chapter 6

Conclusion and Future Works

This thesis has focused on the challenges of acoustic modeling for under-resourced languages. In particular, three novel methods were proposed to improve speech recognition performance under limited training data conditions. The methods are: context-dependent phone mapping, deep neural networks for phone mapping, and exemplar-based acoustic modeling. Experimental results showed that these three methods can work effectively. The contributions of the thesis and possible future research directions will be presented in the following sections.

6.1 Contributions

6.1.1 Context-dependent phone mapping

In chapter 3, a study on the application of cross-lingual phone mapping for under-resourced acoustic modeling using well-trained acoustic models of other languages was presented. This method is motivated by the fact that all human languages share similar acoustic space and therefore most sound units such as phones may be shared by different languages. In our experiments, the source acoustic models were used as a feature extractor to generate cross-lingual features such as source language phone posterior or likelihood scores. These scores were then mapped to the phones of the target language. Different to previous works which used context-independent monophone mapping, this thesis proposed higher mapping resolution by using context-dependent triphone states as the acoustic units for mapping. Experiments conducted in the Wall Street Journal

(WSJ0) corpus verified the effectiveness of the proposed context-dependent phone mapping for building LVCSR models. There are two advantages in the proposed methods, i.e. the use of triphone states to improve acoustic resolution of both source and target models, and the ability to apply different types of source acoustic models. In addition, we found that using nonlinear phone mapping with discriminative training criterion achieves better performance than using linear combination with generative training criterion. The work also showed that the use of different source languages in combination can significantly improve the performance of phone mapping. This work has been published in the journal: IEICE Transactions on Information and Systems [26], and in the two conferences: IALP 2012 [27] and ISCSLP 2012 [28].

6.1.2 Deep neural networks for phone mapping

In the second work described in Chapter 4, the focus was to improve the cross-lingual feature used for mapping, i.e. source language posteriors using deep neural networks (DNNs). Two approaches were investigated. In the first approach, DNNs were used as the source language acoustic model. This was motivated by the fact that DNNs have been shown to have better acoustic modeling than shallow models such as MLPs or GMMs and hence we expected better performance for the phone mapping. The experimental results on the WSJ0 corpus verified that using DNN source language acoustic models produced significantly better results than using shallow source language acoustic models in the cross-lingual phone mapping task. This suggests that the posteriors generated from deep models are of better quality than those generated from shallow models. In addition, we also found that using cross-lingual posteriors as in the phone mapping method is a better choice than using cross-lingual bottleneck feature for the target language acoustic models. In the second approach, DNNs were used to realize the phone mapping as opposed to the conventional MLP approach. Unlike the first approach, using DNNs for the phone mapping is only useful when sufficient target training data are available. This work has been published in the two conferences: APSIPA ASC 2011 [29] and INTERSPEECH 2013 [30].

6.1.3 Exemplar-based acoustic model

The third contribution presented in Chapter 5 proposed a novel exemplar-based acoustic model for limited training data conditions. Exemplar-based model is non-parametric and uses the training samples directly without estimating model parameters. This makes the approach attractive under limited training data conditions. In this study, a specific exemplar-based model called kernel density is used for acoustic modeling. To improve performance of the kernel density acoustic model under limited training data conditions, three approaches have been proposed. In the first approach, a robust cross-lingual bottleneck feature generated by a source language bottleneck neural network was used as the acoustic feature in place of MFCC for the kernel density model of the target language. In the second approach, distance metric learning was applied to learn the distance metric between a test vector and a training exemplar to improve speech recognition performance. Specifically, a Mahalanobis based distance is learnt in an iterative manner to optimize the frame classification accuracy on the training set by maximizing the mutual information criterion (MMI). In the third approach, we proposed a discriminative score tuning module on the top of the kernel density model to fine tune the likelihood scores generated by the kernel density model with a discriminative criterion. Experiments conducted in the WSJ0 corpus showed that with three proposed approaches, performance of the kernel density model is significantly improved and outperforms other acoustic models such as HMM/GMM and even HMM/DNN for limited training data conditions. This work has been published in the conference: INTERSPEECH 2014 [31].

6.2 Future Directions

The objective of the thesis has been to develop novel algorithms to improve speech recognition performance under limited training data conditions. This has resulted in the development of several techniques to address this issue. However, the proposed techniques have several constraints e.g. both training and test data are in clean condition. To generalize the proposed algorithms, the following future research directions are proposed:

Cross-lingual adaptation techniques for phone mapping: Experiments conducted in this thesis used both the training and test sets in clean condition. Our pre-

liminary experiments showed that cross-lingual phone mapping performance drop significantly if there is a mismatch between training and test conditions or between the source language corpus and the target language corpus. To solve this, adaptation techniques can be applied.

Measure the similarity between the source and target acoustic spaces: In this thesis, Malay and Hungarian were used as the source languages to map to English target language. Experimental results showed that different source languages have different effects on phone mapping performance. Hence, in the case that multiple source languages are available, the selection of appropriate source languages to map to the target language is an important question. For the future work, we suggest to find a measure to determine the similarity between two languages' acoustic spaces and discuss how it can affect cross-lingual phone mapping performance. This is not an easy task since not only phonetic overlap between the source and target languages need to be considered but other factors such as corpus recording condition, speaking style, corpus size should also need to be taken into consideration.

Using wider context input: In this thesis, the phone mapping and exemplar models used only one frame of cross-lingual posteriors or bottleneck feature as the input. Recently, with the development of DNN acoustic models, using multi-frames as the input feature to capture broader context is a preferred choice and results in a significant improvement. However, under limited training data conditions, it is not easy to apply this technique since using wider contexts also means increasing the input feature size for target acoustic models. One direction for future work is to find a robust way to use wider contexts while preventing the model from over-fitting.

Pruning in the exemplar-based model: One issue of using the kernel density model is its high computational cost in the proposed distance metric learning and the decoding phases especially when training data increase. To reduce the training and decoding time, pruning techniques will be investigated in the future works.

Using a real under-resourced corpus: In this thesis, English corpus has been chosen as the under-resourced language as the effect of sufficient training data for it is well known. Hence we can focus on the proposed work under the reduced training corpus. In the future, a real under-resourced language should be considered.

Appendix A

Appendix

A.1 Derivation for Distance Metric Learning

In this work, for each input feature $\mathbf{o}_t$, the cost function f which is a function of transformation matrix $\mathbf{Q}$ is defined as the MMI (Maximum Mutual Information) criterion as follows:

$$f(\mathbf{Q}) = \log(p(s_C|\mathbf{o}_t)) = \log\left(\frac{p(\mathbf{o}_t|s_C)p(s_C)}{\sum_{j=1}^J p(\mathbf{o}_t|s_j)p(s_j)}\right) \quad (\text{A.1})$$

where J is the number of states, s_C is the correct state label for input vector $\mathbf{o}_t$, $p(s_C), p(s_j)$ are the state prior estimated from training data, $p(\mathbf{o}_t|s_k)$ with $\forall k \in \{i, C\}$ is the likelihood probability estimated by the kernel density model as in Eq. (A.2) for input feature $\mathbf{o}_t$ and state s_k .

$$p(\mathbf{o}_t|s_k) = \frac{1}{N_k Z} \sum_{i=1}^{N_k} \exp\left(-(\mathbf{Q}\mathbf{o}_t - \mathbf{Q}\mathbf{e}_{ik})^T(\mathbf{Q}\mathbf{o}_t - \mathbf{Q}\mathbf{e}_{ik})\right) \quad (\text{A.2})$$

where $\mathbf{e}_{ik}$ is the i^{th} exemplar of class k , N_k is the number of exemplars in class k , and Z , which is a function of $\mathbf{Q}$, is a normalization term to make Eq. (A.2) be a valid distribution.

Denote

$$g_{ik} = \exp\left(-(\mathbf{Q}(\mathbf{o}_t - \mathbf{e}_{ik}))^T(\mathbf{Q}(\mathbf{o}_t - \mathbf{e}_{ik}))\right) \quad (\text{A.3})$$

we then have

$$p(\mathbf{o}_t|s_k) = \frac{1}{N_k Z} \sum_{i=1}^{N_k} g_{ik}. \quad (\text{A.4})$$

The goal of distance metric learning is to find transformation $\mathbf{Q}$ to maximize f . In this study, the gradient descent method is used. From Eq. (A.1), the derivative of f with respect to $\mathbf{Q}$ is derived as

$$\frac{\partial f}{\partial \mathbf{Q}} = \frac{\partial \log(p(\mathbf{o}_t|s_C)p(s_C))}{\partial \mathbf{Q}} - \frac{\partial}{\partial \mathbf{Q}} \log \sum_{j=1}^J (p(\mathbf{o}_t|s_j)p(s_j)) \quad (\text{A.5})$$

$$= \frac{\partial \log(p(\mathbf{o}_t|s_C)p(s_C))}{\partial \mathbf{Q}} - \frac{1}{\sum_{j'=1}^J (p(\mathbf{o}_t|s_{j'})p(s_{j'}))} \sum_{j=1}^J \frac{\partial (p(\mathbf{o}_t|s_j)p(s_j))}{\partial \mathbf{Q}} \quad (\text{A.6})$$

$$= \frac{\partial \log(p(\mathbf{o}_t|s_C))}{\partial \mathbf{Q}} - \frac{1}{\sum_{j'=1}^J (p(\mathbf{o}_t|s_{j'})p(s_{j'}))} \sum_{j=1}^J \left(p(\mathbf{o}_t|s_j)p(s_j) \frac{\partial \log(p(\mathbf{o}_t|s_j)p(s_j))}{\partial \mathbf{Q}} \right) \quad (\text{A.7})$$

$$= \frac{\partial \log(p(\mathbf{o}_t|s_C))}{\partial \mathbf{Q}} - \sum_{j=1}^J \left(\frac{p(\mathbf{o}_t|s_j)p(s_j)}{\sum_{j'=1}^J (p(\mathbf{o}_t|s_{j'})p(s_{j'}))} \frac{\partial \log(p(\mathbf{o}_t|s_j))}{\partial \mathbf{Q}} \right) \quad (\text{A.8})$$

$$= \frac{\partial \log(p(\mathbf{o}_t|s_C))}{\partial \mathbf{Q}} - \sum_{j=1}^J \left(p(s_j|\mathbf{o}_t) \frac{\partial \log(p(\mathbf{o}_t|s_j))}{\partial \mathbf{Q}} \right) \quad (\text{A.9})$$

where $p(s_j|\mathbf{o}_t)$ is the state posterior.

As $p(\mathbf{o}_t|s_k)$ is defined in Eq. (A.4), we then have

$$\frac{\partial \log(p(\mathbf{o}_t|s_k))}{\partial \mathbf{Q}} = \frac{\partial}{\partial \mathbf{Q}} \left(-\log(N_k) - \log(Z) + \log\left(\sum_{i=1}^{N_k} g_{ik}\right) \right) \quad (\text{A.10})$$

$$= -\frac{\partial}{\partial \mathbf{Q}} \log(Z) + \frac{\partial}{\partial \mathbf{Q}} \log\left(\sum_{i=1}^{N_k} g_{ik}\right). \quad (\text{A.11})$$

By substituting Eq. (A.11) into Eq. (A.9) with $k \in \{i, C\}$, we obtain

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{Q}} = & -\frac{\partial}{\partial \mathbf{Q}} \log(Z) + \frac{\partial}{\partial \mathbf{Q}} \log\left(\sum_{i=1}^{N_C} g_{iC}\right) \\ & - \sum_{j=1}^J \left(p(s_j | \mathbf{o}_t) \left(-\frac{\partial}{\partial \mathbf{Q}} \log(Z) + \frac{\partial}{\partial \mathbf{Q}} \log\left(\sum_{i=1}^{N_j} g_{ij}\right) \right) \right) \end{aligned} \quad (\text{A.12})$$

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{Q}} = & -\frac{\partial}{\partial \mathbf{Q}} \log(Z) + \frac{\partial}{\partial \mathbf{Q}} \log\left(\sum_{i=1}^{N_C} g_{iC}\right) \\ & + \left(\sum_{j=1}^J p(s_j | \mathbf{o}_t) \right) \frac{\partial}{\partial \mathbf{Q}} \log(Z) - \sum_{j=1}^J \left(p(s_j | \mathbf{o}_t) \frac{\partial}{\partial \mathbf{Q}} \log\left(\sum_{i=1}^{N_j} g_{ij}\right) \right). \end{aligned} \quad (\text{A.13})$$

Note that the sum of state posteriors for all states: $\sum_{j=1}^J p(s_j | \mathbf{o}_t) = 1$, and hence normalization term Z in Eq. (A.13) is eliminated. Thus, Eq. (A.13) becomes:

$$\frac{\partial f}{\partial \mathbf{Q}} = \frac{\partial}{\partial \mathbf{Q}} \log\left(\sum_{i=1}^{N_C} g_{iC}\right) - \sum_{j=1}^J \left(p(s_j | \mathbf{o}_t) \frac{\partial}{\partial \mathbf{Q}} \log\left(\sum_{i=1}^{N_j} g_{ij}\right) \right). \quad (\text{A.14})$$

Recall that g_{ik} is defined as in Eq. (A.3) with $k \in \{i, C\}$, we then have

$$\frac{\partial}{\partial \mathbf{Q}} \log\left(\sum_{i=1}^{N_k} g_{ik}\right) = \frac{1}{\sum_{i'=1}^{N_k} g_{i'k}} \sum_{i=1}^{N_k} \frac{\partial g_{ik}}{\partial \mathbf{Q}} \quad (\text{A.15})$$

$$= \frac{1}{\sum_{i'=1}^{N_k} g_{i'k}} \sum_{i=1}^{N_k} \left(g_{ik} \frac{\partial \left(-(\mathbf{o}_t - \mathbf{e}_{ik})^T \mathbf{Q}^T \mathbf{Q} (\mathbf{o}_t - \mathbf{e}_{ik}) \right)}{\partial \mathbf{Q}} \right) \quad (\text{A.16})$$

$$= -\frac{1}{\sum_{i'=1}^{N_k} g_{i'k}} \sum_{i=1}^{N_k} \left(2g_{ik} \mathbf{Q} (\mathbf{o}_t - \mathbf{e}_{ik}) (\mathbf{o}_t - \mathbf{e}_{ik})^T \right) \quad (\text{A.17})$$

$$= -2\mathbf{Q} \sum_{i=1}^{N_k} \left(\frac{g_{ik}}{\sum_{i'=1}^{N_k} g_{i'k}} (\mathbf{o}_t - \mathbf{e}_{ik}) (\mathbf{o}_t - \mathbf{e}_{ik})^T \right) \quad (\text{A.18})$$

$$= -2\mathbf{Q} \sum_{i=1}^{N_k} \left(p_k(i | \mathbf{o}_t) (\mathbf{o}_t - \mathbf{e}_{ik}) (\mathbf{o}_t - \mathbf{e}_{ik})^T \right) \quad (\text{A.19})$$

where $p_k(i|\mathbf{o}_t) = g_{ik} / \sum_{i'=1}^{N_k} g_{i'k}$. If we consider feature distribution $p(\mathbf{o}_t|s_k)$ estimated by the kernel density as a GMM with N_k number of mixtures, $p_k(i|\mathbf{o}_t)$ is then the posterior probability of mixture i^{th} given feature vector $\mathbf{o}_t$ and state s_k .

In summary,

$$\frac{\partial f}{\partial \mathbf{Q}} = 2\mathbf{Q} \left(\sum_{j=1}^J \left(p(s_j|\mathbf{o}_t) \sum_{i=1}^{N_j} \left(p_j(i|\mathbf{o}_t) (\mathbf{o}_t - \mathbf{e}_{ij}) (\mathbf{o}_t - \mathbf{e}_{ij})^T \right) \right) - \sum_{i=1}^{N_C} \left(p_C(i|\mathbf{o}_t) (\mathbf{o}_t - \mathbf{e}_{iC}) (\mathbf{o}_t - \mathbf{e}_{iC})^T \right) \right) \quad (\text{A.20})$$

where with $\forall k \in \{i, C\}$

$$p_k(i|\mathbf{o}_t) = \frac{g_{ik}}{\sum_{i'=1}^{N_k} g_{i'k}} \quad (\text{A.21})$$

$$g_{ik} = \exp \left(-(\mathbf{Q}(\mathbf{o}_t - \mathbf{e}_{ik}))^T (\mathbf{Q}(\mathbf{o}_t - \mathbf{e}_{ik})) \right) \quad (\text{A.22})$$

$$p(s_j|\mathbf{o}_t) = \frac{p(\mathbf{o}_t|s_j)p(s_j)}{\sum_{j'=1}^J (p(\mathbf{o}_t|s_{j'})p(s_{j'}))} \quad (\text{A.23})$$

with

$$p(\mathbf{o}_t|s_j) = \frac{1}{N_j Z} \sum_{i=1}^{N_j} g_{ij}. \quad (\text{A.24})$$

Note that we do not need to compute Z in Eq. (A.24) as after substituting $p(\mathbf{o}_t|s_j)$ from Eq. (A.24) into Eq. (A.23), Z is eliminated.

References

- [1] F. Diehl, *Multilingual and crosslingual acoustic modelling for automatic speech recognition*. PhD thesis, Universitat Politècnica de Catalunya, 2007.
- [2] T. L. Martin, *Towards improved speech recognition for resource poor languages*. PhD thesis, Queensland University of Technology, 2006.
- [3] J. Kohler, “Multilingual phone models for vocabulary-independent speech recognition tasks,” *Speech Communication*, vol. 35, no. 1, pp. 21–30, 2001.
- [4] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, and D. Povey, “Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4334–4337, 2010.
- [5] K. Kirchhoff, “Combining articulatory and acoustic information for speech recognition in noisy and reverberant environments,” in *Proc. International Conference on Spoken Language Processing (ICSLP)*, 1998.
- [6] X. Huang, A. Acero, and H.-W. Hon, *Spoken language processing: a guide to theory, algorithm, and system development*. Prentice Hall, 2001.
- [7] T. Schultz and K. Kirchhoff, *Multilingual speech processing*. Elsevier, Academic Press, 2006.
- [8] T. Schultz and A. Waibel, “Experiments on cross-language acoustic modeling,” in *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 2721–2724, 2001.

- [9] T. Schultz and A. Waibel, “Language-independent and language-adaptive acoustic modeling for speech recognition,” *Speech Communication*, vol. 35, no. 1, pp. 31–51, 2007.
- [10] T. Niesler, “Language-dependent state clustering for multilingual acoustic modelling,” *Speech Communication*, vol. 49, no. 6, pp. 453–463, 2007.
- [11] L. Lu, A. Ghoshal, and S. Renals, “Regularized subspace Gaussian mixture models for speech recognition,” *IEEE Signal Processing Letters*, vol. 18.7, pp. 419–422, 2011.
- [12] L. Lu, A. Ghoshal, and S. Renals, “Maximum a posteriori adaptation of subspace Gaussian mixture models for cross-lingual speech recognition,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4877–4880, 2012.
- [13] A. Stolcke, F. Grezl, M. Hwang, X. Lei, N. Morgan, and D. Vergyri, “Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 321–324, 2006.
- [14] S. Thomas, S. Ganapathy, and H. Hermansky, “Cross-lingual and multistream posterior features for low resource LVCSR systems,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 877–880, 2010.
- [15] P. Lal, *Cross-lingual Automatic Speech Recognition using Tandem Features*. PhD thesis, The University of Edinburgh, 2011.
- [16] K. Vesely, M. Karafiat, F. Grezl, M. Janda, and E. Egorova, “The language-independent bottleneck features,” in *Proc. Spoken Language Technology Workshop (SLT)*, pp. 336–341, 2012.
- [17] N. T. Vu, F. Metze, and T. Schultz, “Multilingual bottle-neck features and its application for under-resourced languages,” in *Proc. International Workshop on*

- Spoken Languages Technologies for Under-resourced Languages (SLTU)*, pp. 90–93, 2012.
- [18] C.-H. Lee, “From knowledge-ignorant to knowledge-rich modeling: a new speech research paradigm for next generation automatic speech recognition,” in *Proc. International Conference on Spoken Language Processing (ICSLP)*, 2004.
- [19] C.-H. Lee, M. A. Clements, S. Dusan, E. Fosler-Lussier, K. Johnson, B. Juang, and L. R. Rabiner, “An overview on Automatic Speech Attribute Transcription (ASAT),” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 513–516, 2007.
- [20] S. Siniscalchi and C.-H. Lee, “A study on integrating acoustic-phonetic information into lattice rescoring for automatic speech recognition,” *Speech Communication*, vol. 51, no. 11, pp. 1139–1153, 2009.
- [21] D. Imseng, H. Bourlard, and P. N. Garner, “Using KL-divergence and multilingual information to improve ASR for under-resourced languages,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4869–4872, 2012.
- [22] P. N. Garner, D. Imseng, J. Dines, H. Bourlard, and P. Motlicek, “Comparing different acoustic modeling techniques for multilingual boosting,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2012.
- [23] K. C. Sim and H. Li, “Context sensitive probabilistic phone mapping model for cross-lingual speech recognition,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2715–2718, 2008.
- [24] K. C. Sim and H. Li, “Stream-based context-sensitive phone mapping for cross-lingual speech recognition,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 3019–3022, 2009.

- [25] K. C. Sim, “Discriminative product-of-expert acoustic mapping for crosslingual phone recognition,” in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 546–551, 2009.
- [26] V. H. Do, X. Xiao, E. S. Chng, and H. Li, “Cross-lingual phone mapping for large vocabulary speech recognition of under-resourced languages,” *IEICE Transactions on Information and Systems*, vol. E97-D, no. 2, pp. 285–295, 2014.
- [27] V. H. Do, X. Xiao, E. S. Chng, and H. Li, “A phone mapping technique for acoustic modeling of under-resourced languages,” in *Proc. International Conference on Asian Language Processing (IALP)*, pp. 233–236, 2012.
- [28] V. H. Do, X. Xiao, E. S. Chng, and H. Li, “Context dependant phone mapping for cross-lingual acoustic modeling,” in *Proc. International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 16–20, 2012.
- [29] V. H. Do, X. Xiao, and E. S. Chng, “Comparison and combination of multilayer perceptrons and deep belief networks in hybrid automatic speech recognition systems,” in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2011.
- [30] V. H. Do, X. Xiao, E. S. Chng, and H. Li, “Context-dependent phone mapping for LVCSR of under-resourced languages,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 500–504, 2013.
- [31] V. H. Do, X. Xiao, E. S. Chng, and H. Li, “Kernel density-based acoustic model with cross-lingual bottleneck features for resource limited LVCSR,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 6–10, 2014.
- [32] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

- [33] H. Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.
- [34] L. R. Rabiner and R. Schafer, *Introduction to digital speech processing*. Foundations and Trends in Signal Processing, 2007.
- [35] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [36] S. Young, “The HTK Book (for HTK Version 3.3),” *Cambridge University Press, Cambridge, UK*, 2005.
- [37] A. Mohamed, G. Dahl, and G. Hinton, “Deep belief networks for phone recognition,” in *Proc. NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.
- [38] A. Mohamed and G. Hinton, “Phone recognition using restricted Boltzmann machines,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4354–4357, 2010.
- [39] G. Dahl, A.-r. Mohamed, and G. E. Hinton, “Phone recognition with the mean-covariance restricted Boltzmann machine,” pp. 469–477, 2010.
- [40] F. Seide, G. Li, X. Chen, and D. Yu, “Feature engineering in context-dependent deep neural networks for conversational speech transcription,” in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 24–29, 2011.
- [41] A. Mohamed, D. Yu, and L. Deng, “Investigation of full-sequence training of deep belief networks for speech recognition,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2010.
- [42] G. Hinton, S. Osindero, and Y. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [43] R. Rosenfeld, “Two decades of statistical language modeling: where do we go from here?,” *Proceedings of the IEEE*, vol. 88, pp. 1270–1278, Aug. 2000.

- [44] H. Ney and S. Ortmanns, “Progress in dynamic programming search for LVCSR,” *Proceedings of the IEEE*, vol. 88, pp. 1224–1240, Aug. 2000.
- [45] X. L. Aubert, “An overview of decoding techniques for large vocabulary continuous speech recognition,” *Computer Speech & Language*, vol. 16, no. 1, pp. 89–114, 2002.
- [46] M. Mohri, F. Pereira, and M. Riley, “Weighted finite-state transducers in speech recognition,” *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.
- [47] W. Byrne, P. Beyerlein, J. M. Huerta, S. Khudanpur, B. Marthi, J. Morgan, N. Petererek, J. Picone, D. Vergyri, and T. Wang, “Towards language independent acoustic modeling,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, pp. 1029–1032, 2000.
- [48] N. Srinivasamurthy and S. Narayanan, “Language-adaptive Persian speech recognition,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2003.
- [49] J. J. Sooful and E. C. Botha, “An acoustic distance measure for automatic cross-language phoneme mapping,” in *Proc. PRASA*, 2001.
- [50] B. Wheatley, K. Kondo, W. Anderson, and T. Muthusamy, “An evaluation of cross-language adaptation for rapid HMM development in a new language,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 237–240, 1994.
- [51] T. Schultz and A. Waibel, “Fast bootstrapping of LVCSR systems with multilingual phoneme sets,” in *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 371–374, 1997.
- [52] J. L. Gauvain and C. H. Lee, “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains,” *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.
- [53] Q. Huo, C. Chan, and C. H. Lee, “Bayesian adaptive learning of the parameters of hidden Markov model for speech recognition,” *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 5, pp. 334–345, 1995.

- [54] C. J. Leggetter and P. C. Woodland, “Flexible speaker adaptation for large vocabulary speech recognition,” *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.
- [55] U. Bub, J. Kohler, and B. Imperl, “In-service adaptation of multilingual hidden-Markov-models,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1451–1454, 1997.
- [56] J. Kohler, “Language adaptation of multilingual phone models for vocabulary independent speech recognition tasks,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 417–420, 1998.
- [57] P. Fung, C. Y. Ma, and W. K. Liu, “Map-based cross-language adaptation augmented by linguistic knowledge: from English to Chinese,” in *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 871–874, 1999.
- [58] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, and A. Rastrow, “The subspace Gaussian mixture model: a structured model for speech recognition,” *Computer Speech & Language*, vol. 25, no. 2, pp. 404–439, 2011.
- [59] L. Lu, A. Ghoshal, and S. Renals, “Regularized subspace Gaussian mixture models for cross-lingual speech recognition,” in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 365–370, 2011.
- [60] H. Hermansky, D. Ellis, and S. Sharma, “Tandem connectionist feature extraction for conventional HMM systems,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1635–1638, 2000.
- [61] F. Grezl, M. Karafiát, S. Kontar, and J. Cernock, “Probabilistic and bottleneck features for LVCSR of meetings,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 757–760, 2007.

- [62] D.-C. Lyu, S. M. Siniscalchi, T.-Y. Kim, and C.-H. Lee, “Continuous phone recognition without target language training data,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2687–2690, 2008.
- [63] S. M. Siniscalchi, D.-C. Lyu, T. Svendsen, and C.-H. Lee, “Experiments on cross-language attribute detection and phone recognition with minimal target-specific training data,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 875–887, 2012.
- [64] H. Bourlard and N. Morgan, “Continuous speech recognition by connectionist statistical methods,” *IEEE Transactions on Neural Networks*, vol. 4, pp. 893–909, 1993.
- [65] V. H. Do, X. Xiao, E. S. Chng, and H. Li, “Context-dependent phone mapping for acoustic modeling of under-resourced languages,” *International Journal of Asian Language Processing*, vol. 23, no. 1, pp. 21–33, 2015.
- [66] V. H. Do, X. Xiao, and E. S. Chng, “Speech attribute recognition using context-dependent modeling,” in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2011.
- [67] G. Aradilla, J. Vepa, and H. Bourlard, “An acoustic model based on Kullback-Leibler divergence for posterior features,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007.
- [68] G. Aradilla, H. Bourlard, and M. Magimai-Doss, “Using KL-based acoustic models in a large vocabulary recognition task,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 928–931, 2008.
- [69] K. C. Sim and H. Li, “Robust phone set mapping using decision tree clustering for cross-lingual phone recognition,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4309–4312, 2008.
- [70] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2000.

- [71] R. Brunelli and T. Poggio, “Face recognition: Features versus templates,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 10, pp. 1042–1052, 1993.
- [72] H. Zhang, A. C. Berg, M. Maire, and J. Malik, “SVM-KNN: Discriminative nearest neighbor classification for visual category recognition,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2126–2136, 2006.
- [73] K. Q. Weinberger, J. Blitzer, and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” in *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 1473–1480, 2005.
- [74] T. Deselaers, G. Heigold, and H. Ney, “Speech recognition with state-based nearest neighbour classifiers,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2093–2096, 2007.
- [75] J. Labiak and K. Livescu, “Nearest neighbor classifiers with learned distances for phonetic frame classification,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2337–2340, 2011.
- [76] N. Singh-Miller and M. Collins, “Learning label embeddings for nearest-neighbor multi-class classification with an application to speech recognition,” in *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 1678–1686, 2009.
- [77] T. N. Sainath, B. Ramabhadran, D. Nahamoo, D. Kanevsky, D. Van Compernelle, K. Demuynck, J. F. Gemmeke, J. R. Bellegarda, and S. Sundaram, “Exemplar-based processing for speech recognition: An overview,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 98–113, 2012.
- [78] J. F. Gemmeke, L. Ten Bosch, L. Boves, and B. Cranen, “Using sparse representations for exemplar based continuous digit recognition,” in *EUSIPCO*, pp. 1755–1759, 2009.

- [79] K. Demuynck, D. Seppi, D. Van Compernelle, P. Nguyen, and G. Zweig, “Integrating meta-information into exemplar-based speech recognition with segmental conditional random fields,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5048–5051, 2011.
- [80] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, New York, 1986.
- [81] X. Xiao, E. S. Chng, T. P. Tan, and H. Li, “Development of a Malay LVCSR System,” in *Proc. Oriental COCODA*, pp. 25–30, 2010.
- [82] V.-B. Le and L. Besacier, “Automatic speech recognition for under-resourced languages: Application to Vietnamese language,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1471–1482, 2009.
- [83] M. J. Gales, “Maximum likelihood linear transformations for HMM-based speech recognition,” *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.
- [84] M.-Y. Hwang and X. Huang, “Shared-distribution hidden markov models for speech recognition,” *Speech and Audio Processing, IEEE Transactions on*, vol. 1, no. 4, pp. 414–420, 1993.
- [85] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.
- [86] X. Li, *Combination and generation of parallel feature streams for improved speech recognition*. PhD thesis, Carnegie Mellon University, 2005.
- [87] X. Cui, J. Xue, B. Xiang, and B. Zhou, “A study of bootstrapping with multiple acoustic features for improved automatic speech recognition,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2009.

- [88] C. Ma, H.-K. Kuo, H. Soltan, X. Cui, U. Chaudhari, L. Mangu, and C.-H. Lee, “A comparative study on system combination schemes for LVCSR,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4394–4397, 2010.
- [89] P. Matejka, P. Schwarz, and J. Cernocky, “Towards lower error rates in phoneme recognition,” in *TSD, Brno, Czech Republic*, 2004.
- [90] A. Abad, T. Pellegrini, I. Trancoso, and J. Neto, “Context dependent modelling approaches for hybrid speech recognizers,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2950–2953, 2010.
- [91] Z. Tuske, M. Sundermeyer, R. Schluter, and H. Ney, “Context-dependent MLPs for LVCSR: Tandem, hybrid or both?,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2012.
- [92] P. Schwarz, P. Matejka, and J. Cernocky, “Hierarchical structures of neural networks for phoneme recognition,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, pp. 1520–6149, 2006.
- [93] V. Nair and G. Hinton, “3-D object recognition with deep belief nets,” *Advances in Neural Information Processing Systems*, no. 22, 2009.
- [94] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, 2006.
- [95] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” *Advances in neural information processing systems*, vol. 19, p. 153, 2007.
- [96] R. Salakhutdinov, *Learning deep generative models*. PhD thesis, University of Toronto, 2009.
- [97] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

- [98] M. Ranzato and G. Hinton, “Modeling pixel means and covariances using factorized third-order boltzmann machines,” in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 2551–2558, IEEE, 2010.
- [99] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.
- [100] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [101] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, “Auto-encoder bottleneck features using deep belief networks,” in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4153–4156, 2012.
- [102] S. Z. Li, “Content-based audio classification and retrieval using the nearest feature line method,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 619–625, 2000.
- [103] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, “Neighbourhood components analysis,” in *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 513–520, 2005.
- [104] D. Povey, *Discriminative training for large vocabulary speech recognition*. PhD thesis, Cambridge University, 2004.
- [105] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.
- [106] X. He and P. Niyogi, “Locality preserving projections,” in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2003.

- [107] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, “A new LDA-based face recognition system which can solve the small sample size problem,” *Pattern recognition*, vol. 33, no. 10, pp. 1713–1726, 2000.