DISTINCT ACOUSTIC MODELING FOR AUTOMATIC SPEECH RECOGNITION

by

KO YU-TING

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
in Computer Science and Engineering

June 2014, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

KO YU-TING

DISTINCT ACOUSTIC MODELING FOR AUTOMATIC SPEECH RECOGNITION

by

KO YU-TING

This is to certify that I have examined the above Ph.D. thesis

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by

the thesis examination committee have been made.

PROF. BRIAN MAK, THESIS SUPERVISOR

PROF. QIONG LUO, ACTING HEAD OF DEPARTMENT

Department of Computer Science and Engineering

5 June 2014

ACKNOWLEDGMENTS

First of all, I would like to thank God for leading me throughout my postgraduate study. Thank him very much for his love, guidance and plans.

I would like to express my sincere thanks to Prof. Brian Mak for his supervision. His teachings and advice are essential in my research work. He taught me not only the knowledge on speech recognition, but also the skills to think, to analyse, to write and to present.

I would like to thank Dr. Manhung Siu for introducing me to Prof. Brian Mak.

I would like to thank members of my PhD thesis examination committee: Prof. Siu-Wing Cheng, Prof. Raymond Wong, Prof. Tan Lee and Prof. Wing-Hung Ki. I would also like to thank Prof. Dit-Yan Yeung for serving as a committee member for my thesis proposal defense.

I would like to express my gratitude to my colleagues Guoli Ye and Dong-Peng Chen. I learnt a lot from them in the past.

Last but not the least, I would also like to thank my mother and my wife for their patience and consistent support. They grant me great freedom to pursue my dream.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DISTINCT ACOUSTIC MODELING FOR AUTOMATIC SPEECH RECOGNITION

by

KO YU-TING

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

ABSTRACT

In triphone-based acoustic modeling, it is difficult to robustly model infrequent triphones due to their lack of training samples. Naive maximum-likelihood (ML) estimation of infrequent triphone models produces poor triphone models and eventually affects the overall performance of an automatic speech recognition (ASR) system. Among different techniques proposed to solve the infrequent triphone problem, the most widely used method in current ASR systems is state tying because of its effectiveness in reducing model size and achieving good recognition results. However, state tying inevitably introduces quantization errors since triphones tied to the same state are not distinguishable in that state. This thesis addresses the problem by the use of distinct acoustic modeling where every modeling unit has a unique model and a distinct acoustic score.

The main contribution of this thesis is the formulation of the estimation of triphone models as an adaptation problem through our proposed distinct acoustic modeling framework named *eigentriphone* modeling. The rational behind *eigentriphone* modeling is that a basis is derived from the frequent triphones and then each triphone is modeled as a point in the space spanned by the basis. The eigenvectors in the basis represent the most important context-dependent characteristics among the triphones

and thus the infrequent triphones can be robustly modeled with few training samples. Furthermore, the proposed framework is very flexible and can be applied to other modeling units. Since grapheme-based modeling is useful in automatic speech recognition of under-resourced languages, we further apply our distinct acoustic modeling framework to estimate context-dependent grapheme models and we call our new method *eigentrigrapheme* modeling. Experimental evaluation of *eigentriphone* modeling was carried out on the Wall Street Journal word recognition task and the TIMIT phoneme recognition task. Experimental evaluation of *eigentrigrapheme* modeling was carried out on four official South African under-resourced languages. It is shown that distinct acoustic modeling using the proposed eigentriphone framework consistently performs better than the conventional tied-state HMMs.

CHAPTER 1

INTRODUCTION

1.1 Automatic Speech Recognition

When we listen to someone talking, we not only receive the speech content, but also identify the language, identity and emotional state of the speaker. Among all these kinds of information, automatic speech recognition (ASR) is aimed at extracting the word sequences transmitted in human speech signals. Fig. 1.1 shows a general structure of an ASR system.

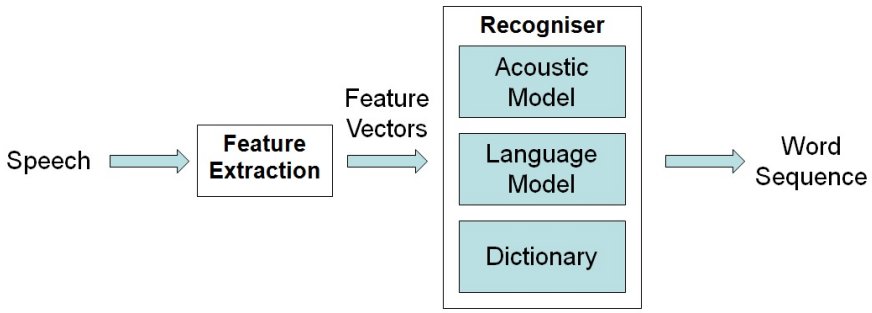

Figure 1.1: General structure of an automatic speech recognition system

First of all, speech signals are converted into sequences of acoustic feature vectors through feature extraction. Feature extraction algorithms are designed to eliminate most of the non-speech variabilities caused by the acoustic conditions such as speakers, recording environment and channels. Then statistical approaches are employed to deal with the speech variabilities in the extracted feature vectors.

Recognition is a search for the word sequence which can best fit the speech data. From a statistical point of view, it is to find a sequence of M words $\hat{W} = w_1, w_2, \ldots, w_M$ that maximizes the posterior probability $P(W|X)$ where $X = x_1, x_2, \ldots, x_T$ is a sequence of T acoustic feature vectors. From the Bayes' rule, we have

$$\hat{W} = \arg\max_W P(W|X) = \arg\max_W \frac{P(W)P(X|W)}{P(X)}.$$

Since $P(X)$ is independent of W, we have

$$\begin{aligned} \hat{W} &= \arg\max_{W} P(X|W)P(W) \\ &= \arg\max_{W} \underbrace{\ln P(X|W)}_{\text{acoustic score}} + \underbrace{\ln P(W)}_{\text{language score}} . \end{aligned} \tag{1.1}$$

Thus, an automatic speech recognition task is formally defined by Eq. (1.1).

From Eq. (1.1), two major components in an ASR system are introduced:

- ***Acoustic model (AM)*** : The acoustic score is computed from a set of acoustic models which describe the statistical behavior of speech in the feature space. The acoustic models consist of a set of hidden Markov models representing each of the basic speech units.

- ***Language model (LM)*** : The language score is computed from the language model which describes the relationship among the co-occurrences of words. The language models normally encapsulate the English grammar information. For example, "IN ORDER" is usually followed by the word "TO". For large vocabulary continuous speech recognition (LVCSR), the language models usually consist of n-grams.

The AM and LM have to be trained before they can be used. Acoustic modeling and language modeling are usually done separately. In this thesis, we focus on acoustic modeling in ASR.

Table 1.1: An example of dictionary used in phone-based ASR systems. The pronunciation of the whole phone set is listed in Table A.1 in Appendix A.

Word	Phonetic Transcription
ABOUT	ah b aw t
CONSIDER	k ah n s ih d er
CAT	k ae t
DOG	d ao g
EAT	iy t
GREEN	g r iy n
HUNDRED	hh ah n d r ah d

If a user wants an ASR system to recognize a particular sentence, he has to define the words appearing in the sentence. For an ASR system, the lexicon and the pronunciation of each word are defined in a dictionary. The pronunciation of each word is defined by listing out its transcription using the basic modeling units. An example of a dictionary used in phone-based ASR systems is shown in Table 1.1.

Phone-based ASR systems refer to using phones as the basic modeling units. In speech science, phonemes are defined as the minimal phonetic units in a language that can distinguish words. For example, there is a phoneme difference in the word pair "DOG" and "FOG" which makes them different. Phones are the acoustic realization of phonemes. In context-independent phone-based modeling, each phone is independently modeled. These phone models are called monophone models. In a typical English ASR system, there are about 40-60 monophones. Although there are other choices of basic units like syllables or words, phone-based modeling is the most popular choice for common ASR systems.

It is observed that the acoustic behavior of a phoneme is highly influenced by its neighbouring phonemes due to coarticulation. For example, the phoneme /t/ sounds differently in the word "UNTAR" (/ah n t aa r/ and "STAR" (/s t aa r/). The phoneme /t/ in the word "STAR" sounds more like the phoneme /d/ because of the influence of its preceding phoneme. Thus, using context-independent models might not enough to cover all the acoustic variations of the phonemes. In 1980, context-dependent phonetic models were proposed [4] and the idea was to replace a single phonetic model by a number of detailed models which are different from one other with different neighbouring units. Context-dependent modeling is much better than context-independent modeling in recognition performance because it covers more acoustic variation by increasing the number of modeling units.

Triphones [82] are the most successful and popular context-dependent modeling units. They are developed from monophones by taking the preceding and following phones into consideration [1]. For example, both models "p-er+t" and "b-er+m" are modeling the phone [er], but they differ from each other with their preceding and following phones. Here, the phone before '-' is the preceding phone and the phone after

[1]For the sake of completeness, there are other context-dependent phone units like biphones and quinphones. The context of a biphone refers its preceding or following phone whereas a quinphone take its neighbouring three phones into consideration

'+' is the following phone.

1.2 Problem of Context-dependent Modeling

During acoustic modeling, since we do not know the sequence of phones in the testing utterances, we have to consider every possible triphone. Thus, if there are N monophones, there will be N^3 triphones altogether. The generation of all possible triphones is called tri-unit expansion. Typically, there are 60,000 - 80,000 triphones. Although using triphones as modeling units can greatly improve the resolution of the acoustic model, the exponential growth of the number of models in the tri-unit expansion brings several drawbacks.

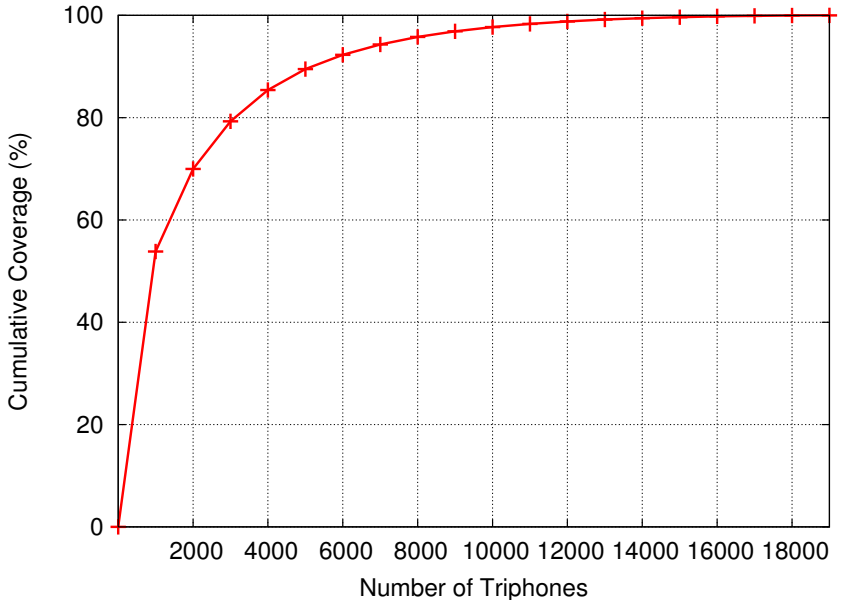

Figure 1.2: Cumulative triphones coverage in the training set of HUB2. The triphones are sorted in descending order of their occurrence count.

First of all, many context-dependent units have insufficient training samples as the amount of training speech data is usually limited. Due to the nature of human speech, the triphones usually distribute very unevenly and most of them do not even appear in the training corpus. For example, Fig. 1.2 depicts the triphone coverage in the HUB2 WSJ0/WSJ1 training corpus [75]. There are 18,991 triphones, and only 3,510 of them have more than 200 samples. That is, about 80% of the training data concentrate on the most common 20% of all seen triphones[2]. Thus a major challenge in context-dependent

[2]Seen triphones are the triphones appearing in the training data. Unseen triphones are the triphones

modeling is to estimate the less frequent context-dependent units reliably, otherwise the poorly trained models may affect the overall performance of an ASR system. On the other hand, a huge increase in model parameters makes heavy demands on the CPU speed and memory size. This made real-time recognition infeasible on many devices, especially embedded devices in the past decades.

Parameter tying has been a common technique used to solve the above problems. The idea is to group the acoustic units of interest into disjoint classes so that members of the same class share the same model parameters and thus their training data. Various parameter tying units have been tried resulting in, for example, generalized triphones [62, 61], tied states [97], shared mixtures [44], and tied subspace Gaussian distributions [8]. However, parameter tying inevitably introduces a quantization error: if two acoustic units are tied together, they become acoustically identical to the speech recognizer. Thus, it has to rely on other constraints such as lexicon or language models to identify the clustered acoustic units and this can potentially harm the discriminative power of the acoustic model.

Since the constraints on CPU speed and memory size are gradually relaxed in the current decade, it is desirable to use more model parameters to achieve better recognition accuracy. If each of the acoustic units is represented by a distinct model, they should be more discriminative. In this thesis, we would like to solve the estimation problem of infrequent triphones by a new distinct acoustic modeling method.

1.3 Distinct Acoustic Modeling

Our investigation on distinct acoustic modeling is motivated by the following two parallel aspects:

- In order to solve the quantization error induced by parameter sharing, we would like to investigate the use of distinct acoustic modeling where every seen triphone has a unique model and a generally distinct acoustic score. The research on distinct acoustic modeling has not been pursued in the past because of limited computing resources but this constraint can be relaxed nowadays.

not appearing in the training data.

- Speaker adaptation techniques [94] have been well developed over the past few decades. Speaker adaptation aims at adapting acoustic models to the characteristics of a particular speaker with a limited amount of speaker specific data. With the success of various speaker adaptation techniques [36, 55, 63], we are motivated to solve the estimation problem of infrequent triphones from an adaptation point of view.

In the past, only a few attempts on distinct acoustic modeling have been made as parameter tying is the mainstream of acoustic modeling. In this thesis, we propose a new distinct acoustic modeling method called *eigentriphone* modeling [52]. Eigentriphone modeling generalizes the idea of eigenvoice speaker adaptation [55] and treats the estimation of infrequent triphones as an adaptation problem. In eigentriphone modeling, a basis is derived over the frequent triphones and each infrequent triphone is modeled as a point in the space spanned by the basis vectors. The eigenvectors in the basis represent the most important context-dependent characteristics among the triphones. By choosing an appropriate number of eigenvectors infrequent triphones can be robustly modeled with few training samples. In contrast to common parameter tying methods, all triphone models are distinct from each other and thus they should be more distinguishable. Experimental evaluations show that using distinct acoustic modeling with our proposed method outperforms the classical state tying method.

We also evaluate another distinct acoustic modeling method named *reference model weighting* [12]. In contrast to eigentriphone modeling, reference model weighting directly uses a set of reference models as the basis. Thus, no eigen-decomposition is required and the training process is faster. Experimental evaluations show that reference model weighting performs as well as eigentriphone modeling and its performance gain is supplementary to the performance of existing state-of-the-art ASR techniques.

Although phone-based modeling is the mainstream in ASR, grapheme-based modeling is popular in under-resourced language ASR [89]. Under-resourced languages refer to languages of which the phonetics and linguistics are not well studied. In this thesis, we also investigate the use of distinct acoustic modeling on grapheme-based ASR systems. We further generalize the eigentriphone modeling framework and apply it to grapheme-based ASR systems. The new method, which we call *eigentrigrapheme* acoustic modeling [51], outperforms the classical grapheme-based modeling method

in several under resourced language recognition tasks.

1.4 Thesis Outline

The organization of this thesis is as follows.

Chapter 2 reviews the fundamental issues of acoustic modeling in ASR with the use of the hidden Markov model, existing parameter reduction techniques, different speaker adaptation schemes and a summary of the past attempts on distinct acoustic modeling.

Chapter 3 is the main part of this thesis. It presents our proposed eigentriphone modeling in detail including the motivation, framework, training procedures and various extensions of our method. Experimental evaluations on both TIMIT and Wall Street Journal corpus are given to show the performance gain of our method over the classical state tying method.

In chapter 4, we investigate the use of distinct acoustic modeling on under-resourced language ASR. We first introduce what under-resourced languages are and then the traditional grapheme-based modeling and our new eigentrigrapheme modeling framework. Experimental evaluation on several under-resourced language recognition tasks are given in this chapter to show the performance gain of our method over the classical grapheme-based modeling.

In chapter 5, we investigate another distinct acoustic modeling method named reference model weighting. Experiments of reference model weighting on Wall Street Journal corpus are implemented to compare its performance with eigentriphone modeling. Then experiments on Switchboard corpus are given to show that the performance gain is supplementary to the performance of existing state-of-the-art ASR techniques.

Chapter 6 concludes the thesis with a summary of contributions and suggestions for future work.

CHAPTER 2

ACOUSTIC MODELING IN SPEECH RECOGNITION

This chapter first gives an introduction to the use of hidden Markov models (HMM) in acoustic modeling in ASR. Then various issues related to acoustic modeling are reviewed including existing parameter reduction techniques, different speaker adaptation schemes and a summary of past attempts in distinct acoustic modeling.

2.1 Hidden Markov Model in ASR

In this section, a review of hidden Markov model (HMM) and phone-based acoustic modeling is given.

For ease of description, let us define:

$\boldsymbol{\lambda}$: an HMM model (normally means all the parameters in the model),

a_{ij}: the transition probability from state i to state j,

J: the total number of states in the HMM $\boldsymbol{\lambda}$,

T: the total number of frames in an observation vector sequence $\mathbf{O}$.

$\mathbf{o}_t$: an observation vector at time t,

$\mathbf{O}$: a sequence of T observation vectors, $[\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T]$,

q_t: the state of $\mathbf{o}_t$ at time t,

$\mathbf{z}$: the state sequence, $[q_1, q_2, \ldots, q_T]$ of $\mathbf{O}$.

The hidden Markov model is a finite state machine. In the case of a continuous HMM, each state is associated with a probability density function (pdf), which is usually a mixture of Gaussians. Transitions among the states are associated with a probability a_{ij} representing the transition probability from state i to state j. HMM is a generative statistical model. In each time step t, the model transits from a source state q_{t-1} to a destination state q_t and an observation vector $\mathbf{o}_t$ is emitted. The distribution of

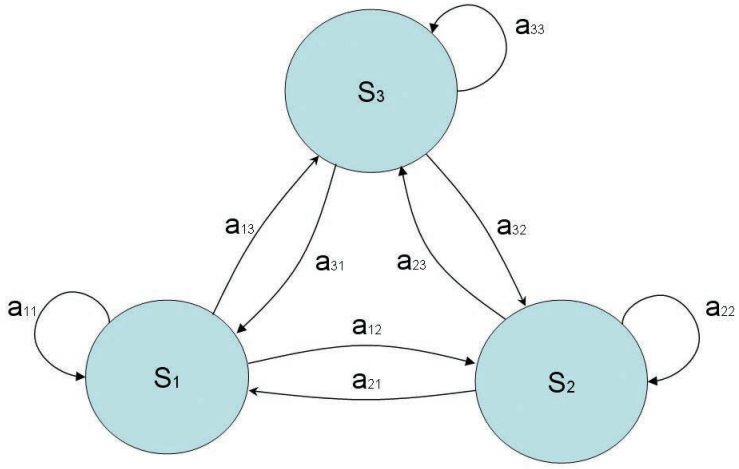

Figure 2.1: An example of HMM with 3 states.

this emitted $\mathbf{o}_t$ is governed by the probability density function in the destination state. The model parameters are the initial probabilities, transition probabilities and the parameters of the set of probability density functions. An example of a first-order HMM is shown in Fig. 2.1.

In a hidden Markov model, the state sequence is not observable whereas only the observations generated by the model are directly visible. The "hidden" Markov model is so named because of the hidden underlying state sequence.

There are three major issues in hidden Markov modeling:

- ***The Evaluation issue*** : From a generative perspective, any sequence of observations of a specified time duration can be generated by a model. Given the HMM parameters $\boldsymbol{\lambda}$, it is possible to determine the probability $P(\mathbf{O}|\boldsymbol{\lambda})$ that a particular sequence of observation vectors $\mathbf{O}$ is generated by the model. In this case, the model parameters $\boldsymbol{\lambda}$ and the observation vector $\mathbf{O}$ are the inputs, and the corresponding probability is the output.

- ***The Training issue*** : From a training/learning perspective, the sequence of observation vectors $\mathbf{O}$ is given whereas the model parameters $\boldsymbol{\lambda}$ are unknown. The observed data gives us some information about the model and we can use them to estimate the model parameters $\boldsymbol{\lambda}$. The given data used for estimation are regarded as the training data. In this case, the observed data $\mathbf{O}$ is the input, and the estimated model parameters $\boldsymbol{\lambda}$ are the outputs.

- ***The Decoding issue*** : In the decoding process, the model parameters $\boldsymbol{\lambda}$ and the sequence of observation vectors $\mathbf{O}$ is given where the sequence of states $\mathbf{z}$ is unknown. The goal is to look for the most likely sequence of underlying states $\mathbf{z}$ which maximizes $P(\mathbf{z}|\mathbf{O}, \boldsymbol{\lambda})$. In this case, the model $\boldsymbol{\lambda}$ and the observation vectors $\mathbf{O}$ are the inputs, and the decoded sequence of states $\mathbf{z}$ is the output.

2.1.1 Assumptions in the Theory of HMM

There are two major assumptions made in the theory of first-order HMMs:

- ***The Markov assumption***: It is assumed that in first-order HMMs the transition probabilities to the next state depend only on the current state and not on the past state history. Given the past k states,

$$P(q_{t+1} = j|q_t = i_1, q_{t-1} = i_2, \ldots, q_{t-k+1} = i_k) = P(q_{t+1} = j|q_t = i_1), \quad (2.1)$$

  where $1 \leq i_1, i_2, \ldots, i_k, j \leq J$.

  On the other hand, the transition probabilities of a k^{th}-order HMM depend on the past k states.

- ***The output independence assumption***: It is assumed that given its emitting state the observation vector is conditionally independent of the past observations as well as the neighbouring states. Hence, we have

$$P(\mathbf{O}|\mathbf{z}, \boldsymbol{\lambda}) = \prod_{t=1}^{T} P(\mathbf{o}_t|q_t, \boldsymbol{\lambda}). \quad (2.2)$$

  If the states are stationary, the observations in a given state are assumed to be independently and identically distributed (i.i.d.).

2.1.2 The Use of HMM as a Phone Model

In phone-based acoustic modeling, the basic modeling units are phones. Each distinct phone in the phone set is modeled by an HMM. The acoustic model consists of a set of phone HMMs. HMMs are used because a speech signal can be viewed as a piecewise stationary signal or a short-time stationary signal.

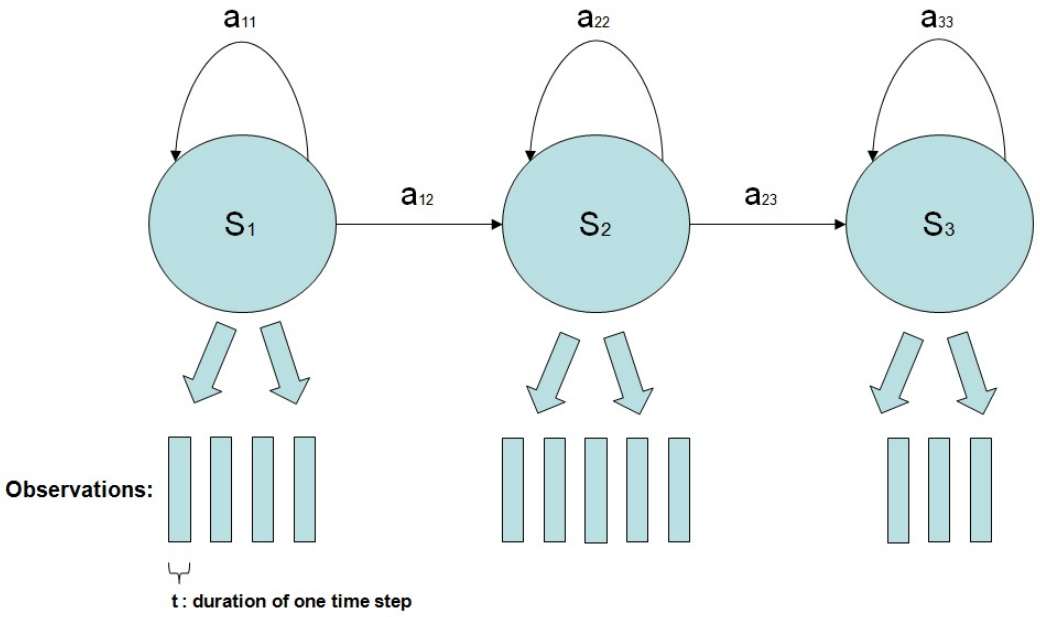

Figure 2.2: An example of a 3-state strictly left-to-right HMM with no skip arcs.

An example of the HMM which is most commonly used to model a phone is shown in Fig. 2.2. and can be treated as a special form derived from the general form in Fig. 2.1 by setting $\{a_{13}, a_{21}, a_{31}, a_{32}\}$ to zero. It is a 3-state strictly left-to-right HMM in which only straight left-to-right transitions are allowed in order to capture the sequential nature of speech. This specific structure makes it easy to connect with another HMM to form a longer HMM. For example, several phone HMMs may connect with each other to form a syllable HMM or a word HMM.

2.1.3 The Choice of Probability Density Function

In Fig. 2.2, the rectangular blocks are the acoustic observations emitted by the HMM state. The statistical behaviour of the emissions is governed by the probability density function (pdf) associated with the states. For the model form of the probability density functions, the Gaussian mixture model (GMM) has been the most common choice in the history of ASR due to its simplicity and trainability. A GMM is a parametric pdf represented as a weighted sum of Gaussian component densities. Given a sufficient number of Gaussian mixtures, GMM could closely approximate any arbitrary continuous density function. With the help of some decorrelating methods [40, 35], diagonal covariance matrices are often used because of their low computational cost.

In this thesis, diagonal covariance matrices are used for every Gaussian component in the acoustic models.

Since the early 90s, the use of artificial neural networks (ANN) [69, 70] to model the emission distributions in HMMs have been proposed. However, compared with the traditional GMM-HMM, little improvement has been made by this ANN-HMM. Due to the limited computing resources in the past, the research on ANN-HMM has not been pursued.

Recently, deep neural network (DNN) [84, 83, 74] is proposed again to model the emission distribution in the HMMs. DNN is conceptually the same as the ANN but differ mainly in the model complexity. The recent DNN is larger in scale than the classical ANN in the following two aspects.

- More output nodes: the number of output nodes is increased from a small number of monophones in ANN to a large number of triphone states in DNN.
- More layers: the number of layers is increased from not more than three layers in ANN to about seven layers in DNN.

Nevertheless, most of the state-of-the-art ASR techniques are developed on the GMM-HMM framework and whether they are feasible on the DNN-HMM framework still needs further investigation. As we would like to compare our proposed method against other ASR techniques, in this thesis we demonstrate our work with implmentations on the GMM-HMM framework.

2.1.4 Training Criteria of HMMs

The HMM parameters are estimated with respect to some objective functions. Maximum likelihood (ML) is the most widely used criterion in HMM training because an efficient training algorithm can be derived. The objective is to find the model parameters that maximise the likelihood of the training data given the correct transcriptions. The standard objective function used in ML training is expressed as

$$F_{ML}(\boldsymbol{\lambda}) = \sum_{r} logP(\mathbf{O}^{(r)}|\hat{h}^{(r)}, \boldsymbol{\lambda}) \tag{2.3}$$

where $\mathbf{O}^{(r)}$ and $\hat{h}^{(r)}$ are the observation vector sequence and the correct hypothesis of the rth utterance respectively. Maximizing the likelihood objective function F_{ML} can be done by the Baum-Welch (BW) algorithm [6, 93] which utilizes the Expectation-Maximization (EM) algorithm.

As mentioned previously, there are several assumptions made in HMM for modeling human speech. These assumptions implies imperfectness of the models and cause the ML training to be suboptimal in terms of recognition accuracy. To address this problem, discriminative training criteria have been proposed as an alternative to the ML criterion. Discriminative training aims to optimize the model parameters such that the recognition error is minimized on the training data. The recognition error is often expressed as different forms of objective functions that involve the correct and the competing hypotheses. Discriminative training has been found to outperform ML training and is widely used in state-of-the-art speech recognition systems. Here, two commonly used discriminative criteria are reviewed.

2.1.4.1 Maximum Mutual Information (MMI)

Maximum mutual information (MMI) [76] criterion aims to optimize the posterior probability, $P(\hat{h}|\mathbf{O}, \boldsymbol{\lambda})$, of the correct transcription given the observation sequence. By applying the Bayes rule, the MMI objective function is expressed as

$$F_{MMI}(\boldsymbol{\lambda}) = \sum_r log \frac{P^k(\mathbf{O}^{(r)}|\hat{h}^{(r)}, \boldsymbol{\lambda}) P^k(\hat{h}^{(r)})}{\sum_h P^k(\mathbf{O}^{(r)}|h^{(r)}, \boldsymbol{\lambda}) P^k(h^{(r)})} \tag{2.4}$$

where $\mathbf{O}^{(r)}$ and $\hat{h}^{(r)}$ are the observation vector sequence and the correct hypothesis of the rth utterance respectively; $h^{(r)}$ in the denominator denotes all possible hypotheses including both the correct and the competing hypotheses; k is empirically used to scale the probability. Although the denominator of eq. 2.4 considers all possible competing hypotheses, in practice, it is approximated by a N-best list [13] which contains the top N competing hypotheses. It is interesting to note that the term $P(\hat{h}^{(r)}|\mathbf{O}^{(r)}, \boldsymbol{\lambda})$ in the numerator of $F_{MMI}(\boldsymbol{\lambda})$ is actually the same as $F_{ML}(\boldsymbol{\lambda})$. Thus, what MMI training is more than ML training is that MMI training maximize the likelihood given the correct transcriptions and at the same time minimize the likelihood given the competing hypotheses.

2.1.4.2 Minimum Phone Error (MPE)

In the MMI objective function, all the competing hypotheses are considered "equal" even though some are better than the others in terms of word error rate (WER) or phone error rate (PER). Thus, it is desirable to incorporate some notion of hypothesis weighting in the discriminative training. Minimum phone error (MPE) [77] is developed to address this problem. The MPE objective function is expressed as

$$F_{MPE}(\boldsymbol{\lambda}) = \sum_{r} log \frac{\sum_{h} P^{k}(\mathbf{O}^{(r)}|h^{(r)}, \boldsymbol{\lambda}) P^{k}(h^{(r)}) A(h^{(r)})}{\sum_{h} P^{k}(\mathbf{O}^{(r)}|h^{(r)}, \boldsymbol{\lambda}) P^{k}(h^{(r)})} \tag{2.5}$$

where $A(h^{(r)})$ represents the weight of hypothesis $h^{(r)}$; $\mathbf{O}^{(r)}$ and $\hat{h}^{(r)}$ are the observation vector sequence and the correct hypothesis of the rth utterance respectively; the index $h^{(r)}$ denotes all possible hypotheses including both the correct and the competing hypotheses or the rth utterance; k is empirically used to scale the probability. From eq. 2.5, we can see that MPE generalize the MMI objective function by replacing the numerator to a sum of all possible hypotheses with $A(h^{(r)})$ associated. If $A(\hat{h}^{(r)}) = 1$ and $A(h^{(r)}) = 0 \ \forall \, h^{(r)} \neq \hat{h}^{(r)}$, the MPE objective is converted back to the MMI objective. In practice, $A(h^{(r)})$ is often rewritten as $A(h^{(r)}, \hat{h}^{(r)})$ to represent a raw phone accuracy for the competing hypothesis $h^{(r)}$ with respect to the correct hypothesis $\hat{h}^{(r)}$.

2.2 Parameter Reduction Techniques

As discussed previously, using context-dependent units can significantly improve the resolution of the acoustic model. The only problem is that trainability becomes a challenge as the number of total units usually grows exponentially. Thus, parameter reduction techniques are proposed to reduce the number of free parameters in the acoustic models. In this section, parameter tying and the most recent canonical state models are reviewed.

2.2.1 Parameter Tying

In the past, different parameter sharing techniques were proposed which could be classified into several categories according to their level of parameter tying [90]. In this

thesis, two typical parameter tying techniques including generalized triphones and state tying are reviewed.

2.2.1.1 Generalized Triphones

As mentioned before, triphones are powerful because they model the most important coarticulatory effects. In the evaluation of triphones, it is observed that some phones have the same effect on their neighboring phones. For example, [b] and [f] have similar effects on the right-neighboring vowel, while [r] and [w] have similar effects on their right-neighboring vowel. Thus, the acoustic behaviour of "b-ae+t" should be similar to "f-ae+t". If these similar triphones can be identified and merged, the number of triphones can be reduced and each model get more training data.

To serve this purpose, generalized triphones [62] were proposed by Kai-Fu Lee. It is a model-based parameter tying method as the whole model, including all the states, is tied to the same cluster. In his paper, he proposed a context merging procedure to identify and merge similar triphone HMMs using the following steps:

Step 1 Generate an HMM for every triphone and train them individually.

Step 2 Create clusters of triphones, with each cluster consisting of one triphone initially.

Step 3 Find the two most similar clusters, and then merge them into one.

Step 4 Go back to Step 3 if the convergence criterion is not met.

One important issue is to define the similarity between two HMMs in step 3. Many similarity measures could be used like cross entropy, divergence and maximum mutual information. In [62], the similarity between two HMMs after merging is defined to be the reciprocal of increased entropy. The more the entropy is increased, the less similar the two HMMs are. The entropy of triphone HMM a is defined as

$$H_a = -\sum_{i=1}^{N_a} P_a(\mathbf{o}_i) log(P_a(\mathbf{o}_i)),$$

where $P_a(o_i)$ is the output probability given the observation vector $\mathbf{o}_i$. If we want to merge triphone HMM a and b into HMM m, the increased entropy can be computed as

$$I(a,b) = (N_a + N_b)H_m - N_a H_a - N_b H_b,$$

where N_a and N_b are the number of training data for model a and model b respectively; H_m is the entropy of the merged model m computed using all the training data of triphone a and b. Experimental results on a 1000-word vocabulary task show that the word accuracy is improved from 95.1% to 95.4% after merging the original 2381 triphones into 1000 generalized triphones.

2.2.1.2 State Tying

Model-based parameter tying is limited in that the left and right contexts cannot be treated independently and hence this inevitably leads to sub-optimal use of the available data. Since coarticulatory effects are more prominent at the onset and ending of a phone than at its center, it will be more flexible if local HMM states can be tied individually instead of the whole triphone HMM. Thus, tied-state HMM (TSHMM) is investigated [97, 96, 79] and experiment results show that state-based clustering consistently out-performed the model-based clustering.

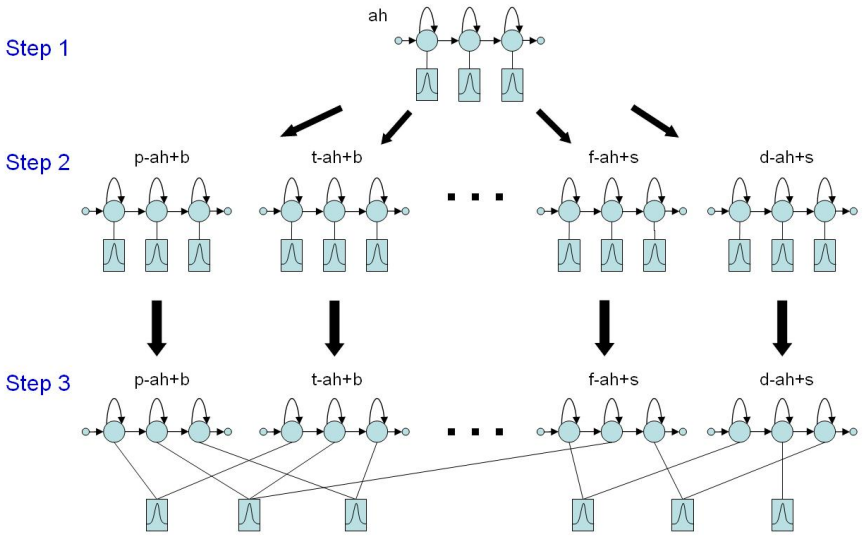

Figure 2.3: The tied-state HMM system building procedure.

A standard procedure of building a tied-state HMM system is illustrated by Fig. 2.3. There are 3 main steps:

Step 1 An initial set of 3-state left to right monophone models is created and trained.

Step 2 These monophone models are then cloned to initialise their corresponding triphone models. Then these triphone models are trained individually.

Step 3 For each set of triphones derived from the same monophone, corresponding states are clustered. For each resulting cluster, its cluster members are tied to the same state and all their training data are used to train that state.

Here, one important issue is to decide upon the clustering mechanism in step 3. There are two common approaches in doing this: the first is a data-driven approach which measure the similarity between states from the training data; the second is a knowledge-based approach which makes use of phonetic knowledge.

2.2.1.3 Data-driven Clustering

The data-driven clustering procedure in state tying is similar to the one used to create generalized triphones. A typical example is senones [45]. Initially all states are placed in individual clusters. The pair of clusters which when combined would form the smallest resultant cluster are merged. This process repeats until either the size of the largest cluster reaches a upper bound or the total number of clusters has fallen below a lower bound. The size of cluster is defined as the greatest distance between any two states. Much the same as in the case of creating generalized triphones, various distance metrics can be used in defining the similarity between states. Practically, the Euclidean distance between the state means scaled by the state variances is usually used [96].

2.2.1.4 Knowledge-based Clustering

One limitation of the data-driven clustering procedure described above is that it does not deal with unseen triphones for which there are no examples in the training data. In 1994, a phonetic knowledge-based clustering method was proposed [79] by Steve Young. In his work, a decision tree which asks phonetic questions about the left and right contexts of each triphone is used. It is shown that tree-based clustering can obtain similar modeling accuracy to that using the data-driven approach but has the additional advantage of providing a mapping for unseen triphones.

A phonetic decision tree is a binary tree in which a yes/no phonetic question is attached to each node. The questions relate to the phonetic context of the triphones. For example, in Fig. 2.4, the question “Is the left neighboring phone of the current triphone a consonant?” is associated with the root node of the tree. Initially all states

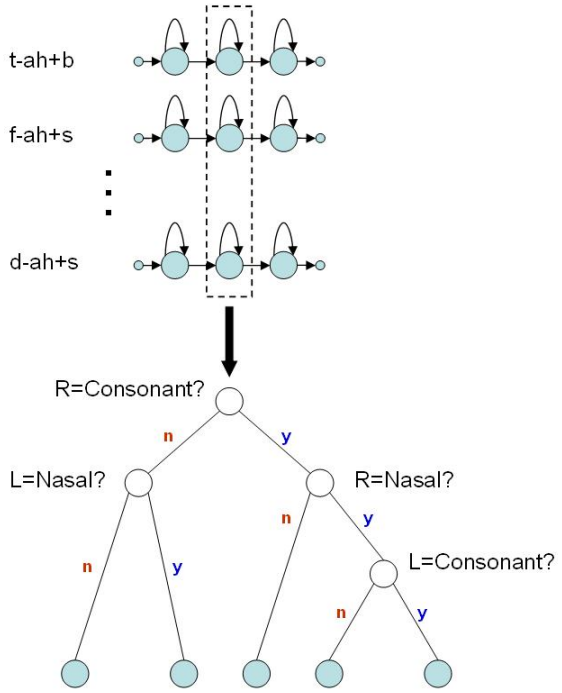

Figure 2.4: Phonetic decision tree-based state tying.

in a given list (typically a specific state position of triphones of the same base phone) are placed at the root node of a tree. Depending on each answer, the pool of states is successively split and this continues until the states have reached the leaf nodes. All states in the same leaf node are then tied. For example, the tree shown in Fig. 2.4 will partition its states into five subsets corresponding to the five terminal nodes. One tree is constructed for each state position of each base phone. The tree topology and questions at each node are chosen to locally maximize the likelihood of the training data and ensure that sufficient data is associated with each tied state. Once all trees have been constructed, unseen triphones can be synthesised by finding the appropriate terminal tree nodes and then using the tied-states associated with those nodes.

Phonetic decision tree-based tying has been the most popular approach in creating acoustic models until now.

2.2.2 Canonical State Models

Among various ways of parameter tying, state tying has been the most popular method for its simplicity and effectiveness. The standard approach for tying the states is to

use phonetic decision trees to determine the sets of tied context-dependent states. Although good performance has been achieved with state tying, the underlying relationship/factor between the context dependent states is not exploited. This motivates the use of a different form of model that attempts to take advantage of this underlying factor. This way of creating context-dependent acoustic models has drawn much attention after the subspace Gaussian Mixture Model (SGMM) [24] was proposed by Daniel Povey in 2010. Afterwards, Mark Gales try to summarize this kind of method by a general framework called the canonical state model (CSM) [33]. To simplify the presentation on SGMM, we first describe the rational behind CSM then the choice of transformation functions that makes SGMM a special case of CSM.

It is assumed in CSMs that every context-dependent states in the system can be transformed from some canonical states. These canonical states represent the underlying factor between the context-dependent states. In standard tying schemes, the model parameters are either independent or identical. In contrast, for a canonical state model, the model parameters are "related" to each other. In other words, a soft tying scheme [31] is being used in CSMs.

In the CSM framework, a canonical state has the form of a standard Gaussian mixture model. Given a canonical state s_g, the likelihood of an observation $\mathbf{o}_t$ at time t is

$$p(\mathbf{o}_t|s_g) = \sum_{m \in s_g} c_g^{(m)} \mathcal{N}(\mathbf{o}_t; \mu_g^{(m)}, \Sigma_g^{(m)}),$$

where $c_g^{(m)}$, $\mu_g^{(m)}$, $\Sigma_g^{(m)}$ are the weight, mean vector and covariance matrix of the mth Gaussian in s_g respectively. Then the context-dependent state s is composed of a mixture of canonical states. The likelihood for context-dependent state s in the CSM framework is given by

$$p(\mathbf{o}_t|s) = \sum_{n=1}^{N} w_s^{(n)} \left(\sum_{m \in s_g} c_s^{(mn)} \mathcal{N}(\mathbf{o}_t; \mu_s^{(mn)}, \Sigma_s^{(mn)}) \right),$$

where N is the number of canonical states and $w_s^{(n)}$ is the weight associated with the nth canonical state. The parameters of state s is generated from the canonical state with the following function:

$$c_s^{(mn)} = F_c(s_g, m; \theta_s^{(n)}),$$

$$\mu_s^{(mn)} = F_\mu(s_g, m; \theta_s^{(n)}),$$
$$\Sigma_s^{(mn)} = F_\Sigma(s_g, m; \theta_s^{(n)}),$$

where $\theta_s^{(n)}$ is the set of transform parameters for component n. Here, it is flexible to define specific transformation functions: F_c, F_μ, and F_Σ for corresponding parameter types.

Canonical state models comprise two sets of parameters: a set of canonical states and a set of transformations. Given that both the canonical state and the transform parameters need to be estimated, the general training process is split into two stages. First the transform parameters are updated given the current canonical state parameters. Second the canonical state parameters are updated given the current transformations.

2.2.2.1 Semi-continuous HMM

Semi-continuous HMM [44] is a special case of CSM. It is the simplest form of CSM as only one single transform component is used. The context-dependent state distribution is given by

$$p(\mathbf{o}_t|s) = \sum_{m \in s_g} c_s^{(m)} \mathcal{N}(\mathbf{o}_t; \mu_s^{(m)}, \Sigma_s^{(m)}).$$

The transformations are defined as

$$F_c(s_g, m; \theta_s^{(n)}) = \frac{\sum_t \gamma_{st}^{(m)}}{\sum_{\tilde{m} \in s_g} \sum_t \gamma_{st}^{(\tilde{m})}},$$
$$F_\mu(s_g, m; \theta_s^{(n)}) = \mu_g^{(m)},$$
$$F_\Sigma(s_g, m; \theta_s^{(n)}) = \Sigma_g^{(m)},$$

where $\gamma_{st}^{(m)}$ is the posterior probability of the mth Gaussian of context-dependent state s generating the observation at time t. We can see from the transformations that the context-dependent state is composed linearly of the Gaussians in the canonical state.

2.2.2.2 Subspace Gaussian Mixture Model (SGMM)

SGMM [24, 22] is a special case of CSM when the transformations are defined as

$$F_c(s_g, m; \theta_s^{(n)}) = \frac{exp(v_g^{(m)T} \theta_s^{(n)})}{\sum_{\tilde{m} \in s_g} exp(v_g^{(\tilde{m})T} \theta_s^{(n)})},$$

$$F_{\mu}(s_g, m; \theta_s^{(n)}) = [\mu_g^{(m1)} \dots \mu_g^{(mP)}]\theta_s^{(n)},$$

$$F_{\Sigma}(s_g, m; \theta_s^{(n)}) = \Sigma_g^{(m)},$$

where $v_g^{(m)}$ is the P-dimensional subspace prior vector for component m and $\theta_s^{(n)}$ is a P-dimensional state-specific vector. The reason it is called a "subspace" model is that the state-specific parameters $\theta_s^{(n)}$ determine the means and weights for all M Gaussian mixtures, which is $M(D+1)$ parameters per state, but the dimension of P will be much less than $M(D+1)$. Thus, the model spans a subspace of the total parameter space. In [24], it is reported that a well-tuned SGMM system will typically have fewer parameters than a well-tuned GMM system, by a factor of two to four. With such a compact model, a smaller amount of training data is sufficient for the training of the state-specific parameters $\theta_s^{(n)}$. This introduces the possibility of training the shared parameters on out-of-domain data and training the state-specific parameters on a smaller amount of in-domain data. With this nice property, SGMM has drawn much attention from the community as now there is a great need of creating ASR systems for new languages (e.g. Arabic). Since collection of training data of a new language usually takes a long time, SGMM can solve the problem by training the shared parameters with existing data (e.g. English training corpus) and training the state-specific parameters with newly collected data. A substantial improvement with the use of SGMM has been reported in a multilingual task [22].

2.3 Speaker Adaptation Techniques

For the training of a large context-dependent acoustic model, speech training data are usually collected from multiple speakers. As a result, the model captures the acoustic variations of different speakers and is known as a speaker-independent (SI) model. An acoustic model that is trained using only speech data from a specific speaker captures only the characteristics of that particular speaker and is known as speaker-dependent (SD) model. Typically, error rates of SI models are two to three times higher than equivalent SD models [60]. However, it is difficult, or sometimes infeasible, to collect a sufficient amount of data from the target speaker. Thus, various speaker adaptation techniques are proposed to adjust the SI models to the characteristics of a target speaker with a limited amount of data. The speaker-adapted (SA) model perform much better

than the SI models for the target speaker.

Existing speaker adaptation techniques can be classified into two main categories: feature-based schemes and model-based schemes. Feature-based schemes aim at transforming the feature vectors whereas model-based schemes aim at modifying the HMM parameters. Since our ultimate goal is to apply these methods to the estimation of infrequent triphones in context-dependent modeling, we focus on the review of model-based adaptation schemes. The most popular model-based adaptation schemes can be categorized into three major families: maximum a posteriori (MAP), linear parameter transformation and speaker-space methods. In this section, the most typical adaptation scheme in each of the above families are introduced. From the literature, most of the error reduction in speaker adaptation came from adapting the mean vector [43]. Thus, in the following we assume that only Gaussian means are adapted.

2.3.1 Maximum A Posteriori (MAP)

MAP adaptation [36, 59, 37] is a Bayesian-based method. It takes advantage of some prior information and adjusts the model parameters based on that information. Let λ be the model parameters and $p(\lambda)$ is the prior probability density function. With the observation data $\mathbf{O}$, the MAP estimate in general is expressed as follows:

$$\begin{aligned} \hat{\boldsymbol{\lambda}} &= \arg\max_{\boldsymbol{\lambda}} P(\boldsymbol{\lambda}|\mathbf{O}) \\ &= \arg\max_{\boldsymbol{\lambda}} P(\mathbf{O}|\boldsymbol{\lambda})P(\boldsymbol{\lambda}) \\ &= \arg\max_{\boldsymbol{\lambda}} logP(\mathbf{O}|\boldsymbol{\lambda}) + logP(\boldsymbol{\lambda}) \end{aligned} \tag{2.6}$$

If there is no prior information about the model parameters, $P(\boldsymbol{\lambda})$ becomes a uniform distribution and the MAP estimate becomes identical to the maximum likelihood (ML) estimate.

In fact, the density function $P(\boldsymbol{\lambda})$ has to be carefully selected so that the maximum a posteriori can be effectively evaluated. If the state observation density is a mixture of Gaussians (GMM), we have [1]

$$P(\mathbf{O}|\boldsymbol{\lambda}) = \sum_{r=1}^{R} w_r \mathcal{N}(\mathbf{O}|u_r, \sigma_r) \tag{2.7}$$

[1] To simplify our presentation here, we assume a mixture of univariate normal densities.

where R is the number of Gaussians; w_r, u_r and σ_r are the mixture weight, mean and variance of the rth Gaussian respectively. As we are going to adapt the Gaussian means only, the prior density is selected as a product of Gaussian distribution by the fact that it is the conjugate distribution of the GMM. Thus, we have

$$P(\lambda) \quad \propto \quad \prod_{r=1}^{R} exp[-\frac{1}{2}(\frac{u_r - u_{0r}}{\sigma_{0r}})^2] \tag{2.8}$$

where u_{0r} and σ_{0r} are the mode and variance of prior density of u_r respectively.

Applying the EM algorithm, we can write the auxiliary function as follows:

$$Q(\lambda) \quad = \quad \sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_r(t)(\frac{o_t - u_r}{\sigma_r})^2 + \sum_{r=1}^{R}(\frac{u_r - u_{0r}}{\sigma_{0r}})^2 \tag{2.9}$$

Take the derivative of each u_r and we arrive at the following solution:

$$\hat{u}_r = \frac{\tau_r u_{0r} + \sum_{t=1}^{T}\gamma_r(t)o_t}{\tau_r + \sum_{t=1}^{T}\gamma_r(t)} \tag{2.10}$$

where $\tau_r = \frac{\sigma_r}{\sigma_{r0}}$ and $\gamma_r(t)$ is the occupation probability of the rth Gaussian given observation x_t.

From eq. (2.10), we can see that the MAP estimate mean $\hat{u}_r$ is actually a weighted sum of the mode of the prior density with the ML estimate mean $\frac{\sum_{t=1}^{T}\gamma_r(t)x_t}{\sum_{t=1}^{T}\gamma_r(t)}$. For a speaker adaptation task, the mode of the prior density u_{0r} can be obtained from the equivalent SI model.

MAP has an advantage that it converges to an SD model when the adaptation data increases. However, its limitation is that only Gaussians that occur in the adaptation data can be modified from the prior SI model. The correlations between model parameters are also not fully utilized.

2.3.2 Maximum Likelihood Linear Regression (MLLR)

MLLR [63] is a transformation-based adaptation method. The mean vector $\hat{\boldsymbol{\mu}}_r$ of the rth Gaussian of the SA model is adapted from the mean vector μ_r of the equivalent SI model as follows:

$$\hat{\boldsymbol{\mu}}_r = \mathbf{A}\boldsymbol{\mu}_r + \mathbf{b} \tag{2.11}$$

where $\mathbf{A}$ is a transformation matrix and $\mathbf{b}$ is a bias. If $\mathbf{A}$ and b are used to transform the mean vectors of every Gaussian, they are called a global transform. In fact, the MLLR adaptation usually groups the Gaussians into several regression classes. The Gaussians in the same regression class share the same transformation matrix and bias. Thus, eq. (2.11) can be generalized into

$$\hat{\boldsymbol{\mu}}_r = \mathbf{A}_c \boldsymbol{\mu}_r + \mathbf{b}_c \tag{2.12}$$

where $\mathbf{A}_c$ is the transformation matrix and $\mathbf{b}_c$ is the bias of regression class c which $\boldsymbol{\mu}_r$ belongs. The transformation parameters are estimated with an ML approach.

The number of free parameters of the MLLR transform can be controlled by the number or regression classes and the choice of transformation matrix such as diagonal matrix, block diagonal matrix or full matrix, usually decided by the amount of adaptation data. With more adaptation data, a more precise transformation can be achieved.

MLLR works well when a certain amount of adaptation data is available. In [65], MLLR outperforms all other methods when it is given 10 seconds of adaptation data.

2.3.2.1 Constrained MLLR

The MLLR transform described previously is also called unconstrained MLLR where the mean vectors and covariance matrices of the Gaussian components are transformed separately or, as described, the covariance matrices remain unchanged. In contrast, applying the same transform to a pair of corresponding mean vector and covariance matrix is referred to as constrained MLLR (CMLLR) [20, 30]. The mean vector $\hat{\boldsymbol{\mu}}_r$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_r$ of the rth Gaussian of the SA model is adapted from the μ_r and Σ_r of equivalent SI model as follows:

$$\hat{\boldsymbol{\mu}}_r = \mathbf{A}' \boldsymbol{\mu}_r + \mathbf{b}' \tag{2.13}$$

$$\hat{\boldsymbol{\Sigma}}_r = \mathbf{A}' \boldsymbol{\Sigma}_r \mathbf{A}'^T \tag{2.14}$$

where $\mathbf{A}'$ is the constrained linear transform and $\mathbf{b}'$ is the bias on the mean vector.

One disadvantage of the above model-based formulation is that the adapted covariance matrix $\hat{\boldsymbol{\Sigma}}_r$ is non-diagonal if A' is non-diagonal and becomes computationally expensive to calculate the likelihood with a non-diagonal covariance matrix. Luckily,

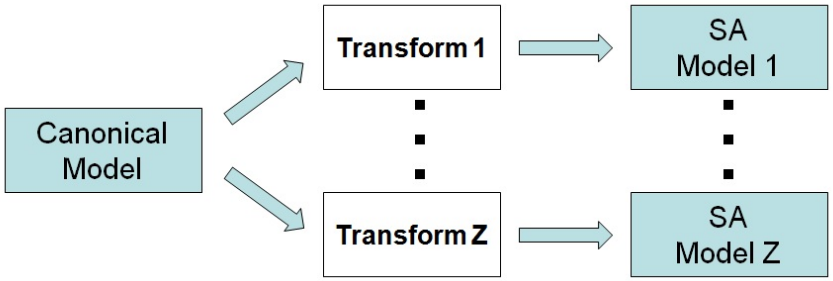

Figure 2.5: Illustration of speaker adaptive training.

this problem can be avoided by rewriting the above formulation into a feature transform. An equivalent log likelihood of an observation o_t given the adapted parameters is computed by:

$$log\mathcal{N}(\mathbf{o}_t; \hat{\boldsymbol{\mu}}_r, \hat{\boldsymbol{\Sigma}}_r) = log\mathcal{N}(\hat{\mathbf{o}}_t; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r) + log|\mathbf{F}| \tag{2.15}$$

where

$$\hat{\mathbf{o}}_t = \mathbf{A}'^{-1}\mathbf{o}_t - \mathbf{A}'^{-1}\mathbf{b}' = \mathbf{F}\mathbf{o}_t - \mathbf{g}, \tag{2.16}$$

$\mathbf{F} = \mathbf{A}'^{-1}$ is the feature transform matrix and $\mathbf{g} = \mathbf{A}'^{-1}\mathbf{b}'$ is the bias. Once the feature is transformed, the likelihood can be computed by the original diagonal covariance matrix. Thus, in practice, CMLLR is usually implemented as a feature transform and is also known as feature MLLR (FMLLR).

2.3.2.2 Speaker Adaptive Training (SAT)

In the MLLR adaptation scheme described previously, the SA model for a target speaker is adapted from an SI model where the SI model is estimated by mixing the data from all the training speakers. This common SI training paradigm does not make use of the characteristics of the individual training speaker and the resulting SI model might not be fitted to the source of adaptation. Speaker adaptive training (SAT) [1] is a framework proposed to improve the quality of the source model.

The SAT framework is illustrated in Fig. 2.5. It is assumed that every SA model, including the training and testing speakers, are transformed from a canonical model. In the training phase, the canonical model and the transforms are estimated to maximize the likelihood of training data given the SA model for each training speaker. Then only

the canonical model is needed to be the source of adaptation during recognition. For the transforms, usually either MLLR or CMLLR is used.

2.3.3 Eigenvoice (EV)

Eigenvoice [55, 27, 57, 56] is an eigenspace-based adaptation method which targets adaptation when the amount of adaptation data is very limited. In the EV approach, a set of T SD models are trained from T training speakers. Then from each SD model, a supervector of dimension D is constructed by concatenating all the Gaussian means in that SD model. After collecting the T supervectors, PCA is applied and only the first K eigenvectors are used. These K eigenvectors, which we called "eigenvoices", capture the most important speaker characteristics from the training speaker. Here, the dimension is greatly reduced as $K < T << D$. The new speaker's supervector, $\mathbf{s}$, is assumed to lie in the speaker-space spanned by these K eigenvoices. Thus, it is represented by a weighted sum of the eigenvoices as follows:

$$\mathbf{s} = \mathbf{e}_0 + \sum_{k=1}^{K} w_k \mathbf{e}_k \tag{2.17}$$

where $\mathbf{e}_0$ is the mean of all supervectors; $\mathbf{e}_k$ and w_k are the kth eigenvoice and its weight. The weights are estimated using the new speaker's adaptation data with an ML approach called maximum likelihood eigen decomposition (MLED).

The main advantage of EV is that it works better than other adaptation methods under a very limited amount of adaptation data. In [65], EV outperforms MLLR when the amount of adaptation data is less than 5 seconds. In an alphabet recognition task [57], EV works better than both MAP and MLLR with only a few letters of adaptation data. The drawback of EV is that its gain with increasing data is limited and in such cases MLLR is a better choice.

2.3.3.1 Reference Speaker Weighting (RSW)

Reference speaker weighting [65, 41] is very similar to Eigenvoice adaptation as it also requires the modeling of a new speaker to lie in a speaker-space. Indeed, they differ only in how the basis is computed. EV computes a set of orthogonal basis vectors

through PCA whereas RSW uses a set of reference speaker vectors as the basis. In [41], the reference speakers are computed through a hierarchical speaker clustering (HSC) algorithm. However, it is reported in [65] that simply using the models of all the training speakers as the reference gives better results.

2.4 Distinct Acoustic Modeling for ASR

While parameter tying has become the main approach in context-dependent acoustic modeling, it has one potential drawback — if two acoustic units are clustered together, they become acoustically identical to the speech recognizer. Thus, it has to rely on other constraints such as lexicon or language models to identify the clustered acoustic units. This motivates the use of distinct acoustic modeling where every context-dependent unit has a unique acoustic score. In this section, we review the past attempts in distinct acoustic modeling.

2.4.1 Attempts in Distinct Acoustic Modeling

Model interpolation [82, 81, 14] is the earliest example of distinct context-dependent modeling. It creates trpihone models by a combination of some reference models. Although these reference models capture weaker contextual information, they are robustly trained. In [81], the parameters of a triphone model are generated by an interpolation of the model itself with some left-context, right-context or context-independent models. Each pdf in a model is given a different weight according to its state position (for example, left-context models have greater weights for pdfs in the leftmost states) and its number of training samples (for example, if a triphone appears many times, its weight will dominate). Although interpolation with better trained models makes triphones usable, the infrequent triphones are still undertrained and lead to a modest performance.

Recently, another attempt at model interpolation named the back-off acoustic modeling [11] has been proposed. In their work, the acoustic score of a triphone is computed from an interpolation between its native model and models based on broad phonetic class contexts. Thus, it can guarantee that every triphone has a distinct acoustic score.

Given a sequence of feature vectors $\mathbf{O}$, let $a(\mathbf{O}, l)$ be the acoustic score returned by the model of triphone l. If there are only a few training samples of triphone l, the score $a(\mathbf{O}, l)$ may not be accurate as the model itself is not robustly estimated. The idea of back-off acoustic models is that they use the back-off score $\tilde{a}(\mathbf{O}, l)$ to replace $a(\mathbf{O}, l)$ so that those inaccurate scores are linearly weighted with some entrusted scores.

Now the triphone label l can generally be replaced by $\langle p_l|p_c|p_r\rangle$ where p_l, p_c and p_r stand for the left phone, current phone and right phone respectively. Let $B(p)$ denote the broad phonetic class of base phone p , where the mapping function $B()$ can be constructed according to some acoustic phonetic properties, such as manner of pronunciation or articulation place. For example, /d/ and /t/ might be assigned to the same phonetic class so that $B(/d/) = B(/t/)$. Given the broad phonetic class assignments, the back-off acoustic score $\tilde{a}(\mathbf{O}, \langle p_l|p_c|p_r\rangle)$ of a triphone $\langle p_l|p_c|p_r\rangle$ can be computed as:

$$\tilde{a}(\mathbf{O}, \langle p_l|p_c|p_r\rangle) = w_0 a(\mathbf{O}, \langle p_l|p_c|p_r\rangle) + w_r \tilde{a}(\mathbf{O}, \langle p_l|p_c|B(p_r)\rangle) + w_l \tilde{a}(\mathbf{O}, \langle B(p_l)|p_c|p_r\rangle),$$

where w_0, w_r and w_l are the linear weights for combining the scores and they should sum up to one. The two back-off scores $\tilde{a}(\mathbf{O}, \langle p_l|p_c|B(p_r)\rangle)$ and $\tilde{a}(\mathbf{O}, \langle B(p_l)|p_c|p_r\rangle)$ can similarly be further decomposed. Here, the weights w_0, w_r and w_l are determined by the amount of training data. For example, the value of w_0 is proportional to the the occurrences of the triphone $\langle p_l|p_c|p_r\rangle$ in the training data. In other words, if the training data of triphone $\langle p_l|p_c|p_r\rangle$ is enough, the value of w_0 will get close to one and thus $\tilde{a}(\mathbf{O}, \langle p_l|p_c|p_r\rangle) = a(\mathbf{O}, \langle p_l|p_c|p_r\rangle)$.

Although improvements are reported in [11], acoustic-phonetic knowledge is required to derive the broad phonetic classes. Thus, the method itself is difficult to port between different phone sets. On the other hand, how to get the "optimal" broad phonetic classes for any modeling units requires further investigation.

CHAPTER 3

EIGENTRIPHONE MODELING

We pointed out the problem of conventional parameter tying methods in Chapter 1 that quantization errors are induced when distinct triphones are tied together and represented by the same model. To address the problem, we investigate the use of distinct acoustic modeling with our proposed method called *eigentriphone* modeling. This chapter starts with the motivation and then the description of *eigentriphone* modeling. There are three variants of the method, namely the model-based, state-based and cluster-based eigentriphone modeling. The three variants differ in the modeling unit (triphones or triphone states) and resolution. For ease of understanding, the basic procedure of model-based eigentriphone modeling is first given, followed by that of the other two variants. Two extensions of our modeling framework: derivation of bases using weighted PCA and estimation of coefficients using penalized maximum likelihood eigen decomposition (PMLED) are also described. After that, experiments on TIMIT phoneme recognition and Wall Street Journal continuous speech recognition are given.

3.1 Motivation from Eigenvoice Adaptation

From an adaptation point of view, our eigentriphone modeling is motivated by eigenvoice adaptation [54] in a way that the estimation of triphones with insufficient training samples is treated as an adaptation problem. Compared with other speaker adaptation techniques, the eigenvoice approach is more appropriate for our task. This is because eigenvoice performs well when the amount of adaptation data is less than 5 seconds and is better than all other methods when there is only 2 seconds of adaptation data. Empirically, we define triphones with less than 3 seconds of training data as infrequent triphones.

The frameworks of eigenvoice adaptation and eigentriphone acoustic modeling are very similar except for the following differences:

- Speaker-dependent models in eigenvoice are replaced by triphone models in eigentriphone modeling. Thus, the dimensionality of the supervectors in eigentriphone modeling should be smaller than that in eigenvoice adaptation.
- There are multiple sets of eigenvectors in eigentriphone modeling whereas there is only one set of eigenvectors in eigenvoice adaptation.
- Usually few speakers are adapted in eigenvoice whereas there are at least thousands of triphones that need to be adapted. In addition, adapted triphones have to work together as a complete acoustic model.

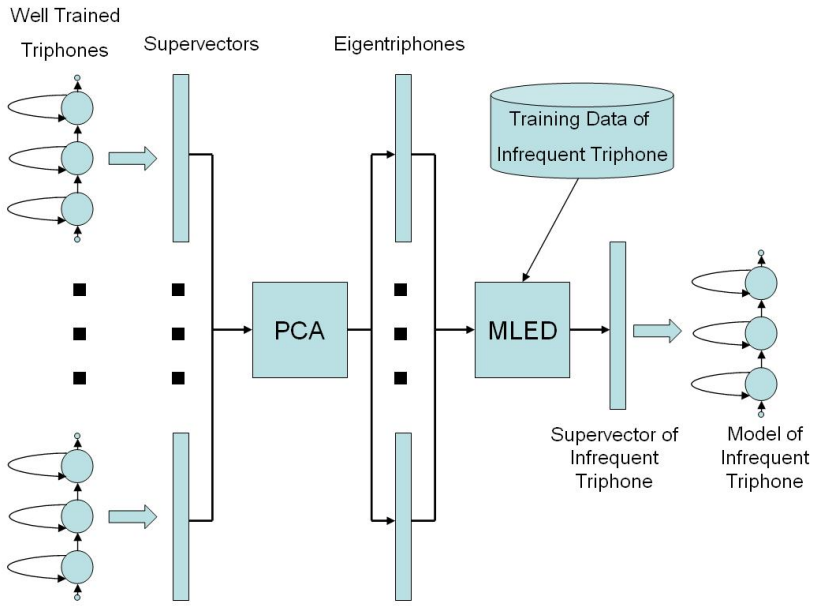

Figure 3.1: The model-based eigentriphone modeling framework.

3.2 The Basic Procedure of Eigentriphone Modeling

In eigentriphone modeling, a set of eigenvectors is derived from the triphones using principle component analysis (PCA). These eigenvectors, which we call *eigentriphones*, capture the most important context-dependent characteristics. Each triphone / triphone state is then modeled as a point in the space spanned by the eigentriphones. By using only the leading eigentriphones, the dimensionality of the new space is greatly reduced compared with the original acoustic space, so that even the infrequent triphones / triphone states can be estimated robustly with few training samples. There are

three variants of the method, namely the model-based, state-based and cluster-based eigentriphone modeling. The three variants differ in the modeling unit (triphones or triphone states) and resolution. For the ease of understanding, the basic procedures of model-based eigentriphone modeling are first given and then the other two variants. To give an idea of the general framework, Fig. 3.1 shows an overview of model-based eigentriphone acoustic modeling.

3.2.1 Model-based Eigentriphone Modeling

In model-based eigentriphone modeling, the whole triphone model, including all the states, is adapted using the same set of eigentriphone coefficients. The supervectors are constructed by concatenating the Gaussian mean vectors from all the states of each triphone HMM of a base phone. The eigenvectors generated by the PCA have the same dimensionality as the supervectors and they construct a "triphone model space". Each triphone model is modeled as a point in this space and the eigentriphone coefficients are estimated by *maximum-likelihood eigen-decomposition* (MLED) [54].

The basic procedure of model-based eigentriphone modeling is described as below. The following procedures are repeated for each base phone i using all its triphones that appeared in the training corpus:

STEP 1 : Monophone hidden Markov model (HMM) of base phone i is first estimated from the training data. Each monophone is a 3-state strictly left-to-right HMM, and each state is represented by an M-component Gaussian mixture model (GMM).

STEP 2 : The monophone HMM of base phone i is then cloned to initialize *all* its N_i triphones in the training data. Note that (a) unlike common triphone cloning from an HMM with 1-component GMM states, in our eigentriphone procedure, triphones are cloned from an monophone HMM with M-component GMM states, and (b) no state tying is performed.

STEP 3 : Re-estimate only the Gaussian means of the triphones after cloning; their Gaussian covariances and mixture weights (which are copied from their base phone HMM) remain unchanged.

STEP 4 : Create a triphone supervector $\mathbf{v}_{ip}$ for each triphone p of base phone i by

stacking up all the Gaussian mean vectors from its three states as below.

$$\mathbf{v}_{ip} = \begin{bmatrix} \boldsymbol{\mu}_{ip11}, & \boldsymbol{\mu}_{ip12}, & \cdots, & \boldsymbol{\mu}_{ip1M}, \\ \boldsymbol{\mu}_{ip21}, & \boldsymbol{\mu}_{ip22}, & \cdots, & \boldsymbol{\mu}_{ip2M}, \\ \boldsymbol{\mu}_{ip31}, & \boldsymbol{\mu}_{ip32}, & \cdots, & \boldsymbol{\mu}_{ip3M} \end{bmatrix} , \tag{3.1}$$

where $\boldsymbol{\mu}_{ipjm}, j = 1, 2, 3$, and $m = 1, 2, \ldots, M$ is the mean vector of the mth Gaussian component at the jth state of triphone p of base phone i. Similarly, a monophone supervector $\mathbf{m}_i$ is created from the monophone model of the base phone i.

STEP 5 : Let N_i be the number of triphones of base phone i. Collect all triphone supervectors $\mathbf{v}_{i1}$, $\mathbf{v}_{i2}$, ..., $\mathbf{v}_{iN_i}$ as well as the monophone supervector $\mathbf{m}_i$ of base phone i, and derive an eigenbasis from their correlation or covariance matrix using *principal component analysis* (PCA). The covariance matrix is computed as follows:

$$\frac{1}{N_i} \sum_p (\mathbf{v}_{ip} - \mathbf{m}_i)(\mathbf{v}_{ip} - \mathbf{m}_i)' . \tag{3.2}$$

Notice that the monophone supervector $\mathbf{m}_i$, instead of the mean of triphone supervectors, is used to "center" triphone supervectors so that in the worst case the poor triphones may fall back to the monophone HMM[1].

STEP 6 : Arrange the eigenvectors $\{\mathbf{e}_{ik}, k = 1, 2, \ldots, N_i\}$ in descending order of their eigenvalues λ_{ik}, and pick the top K_i (where $K_i \leq N_i$) eigenvectors to represent the eigenspace of base phone i. These K_i eigenvectors are now called *eigentriphones* of phone i. In general, different base phones have different numbers of eigentriphones, depending on the criterion used to decide the value of K_i.

STEP 7 : Now the supervector $\mathbf{v}_{ip}$ of any triphone p of base phone i is assumed to lie in the space spanned by the K_i eigentriphones. Thus, we have

$$\mathbf{v}_{ip} = \mathbf{m}_i + \sum_{k=1}^{K_i} w_{ipk} \mathbf{e}_{ik} , \tag{3.3}$$

where $\mathbf{w}_{ip} = [w_{ip1}, w_{ip2}, \ldots, w_{ipK_i}]$ is the eigentriphone coefficient vector of triphone p in the "triphone space" of base phone i.

[1]Empirically, we find that centering by the monophone supervector gives slightly better performance than if the mean of the triphone supervectors is used.

STEP 8 : Estimate the eigentriphone coefficient vector $\mathbf{w}_{ip}$ of any triphone p by maximizing the likelihood $L(\mathbf{w}_{ip})$ of its training data:

$$L(\mathbf{w}_{ip}) = \text{constant} - \sum_{j,m,t} \gamma_{ipjm}(t)(\mathbf{x}_t - \boldsymbol{\mu}_{ipjm}(\mathbf{w}_{ip}))' C_{ipjm}^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_{ipjm}(\mathbf{w}_{ip})) \tag{3.4}$$

where C_{ipjm} and $\gamma_{ipjm}(t)$ are the covariance and occupation probability of the mth Gaussian at the jth state of triphone p of base phone i given observation $\mathbf{x}_t$. The procedure is called *maximum-likelihood eigen-decomposition* (MLED) in [54]. Finally, the Gaussian mean of the mth mixture at the jth state of triphone p can be obtained from $\mathbf{v}_{ip}$ as

$$\boldsymbol{\mu}_{ipjm} = \mathbf{m}_{ijm} + \sum_{k=1}^{K_i} w_{ipk}\mathbf{e}_{ikjm} \ . \tag{3.5}$$

STEP 9 : If either the eigentriphone coefficients converge or the recognition accuracy of a development data set is maximized, go to STEP 10. Otherwise, re-align the training data using the model in STEP 8, re-estimate the Gaussian means and repeat STEP 4 – 9.

STEP 10 : After the eigentriphone "adaptation" of the Gaussian means, the Gaussian covariances and mixture weights of a triphone are re-estimated if its sample count is greater than the thresholds θ_v and θ_w respectively. Otherwise, they remain the same as those of the monophone model from which they are cloned.

3.2.2 State-based Eigentriphone

In model-based eigentriphone acoustic modeling, high-dimensional triphone supervectors are constructed by concatenating the Gaussian mean vectors from all the (three) states of each triphone HMM of a base phone. One may also apply the modeling framework to sub-phonetic units. Motivated by the findings that the correlation between states is usually stronger than the correlation between the entire HMMs [79], *state-based eigentriphone* modeling was also investigated. Now, an eigenbasis is developed for each state of each base phone in a procedure similar to that of model-based eigentriphone modeling in Section 3.2.1. Compared with model-based eigentriphone

acoustic modeling, state-based eigentriphone acoustic modeling produces three times more eigenbases as well as the eigentriphone coefficients, but its eigenvector dimension is only $1/3$ of the former.

The basic procedure of state-based eigentriphone modeling is described as below. The following procedures are repeated for each state j of each base phone i:

STEP 1 : Monophone hidden Markov model (HMM) of base phone i is first estimated from the training data. Each monophone is a 3-state strictly left-to-right HMM, and each state is represented by an M-component Gaussian mixture model (GMM).

STEP 2 : The monophone HMM of base phone i is then cloned to initialize *all* its N_i triphones in the training data.

STEP 3 : Re-estimate only the Gaussian means of the triphones after cloning; their Gaussian covariances and mixture weights (which are copied from their base phone HMM) remain unchanged.

STEP 4 : Create a triphone state supervector $\mathbf{v}_{ijp}$ for state j of triphone p of base phone i by stacking up all the Gaussian mean vectors as below.

$$\mathbf{v}_{ijp} = \left[\boldsymbol{\mu}_{ipj1}, \quad \boldsymbol{\mu}_{ipj2}, \quad \cdots, \quad \boldsymbol{\mu}_{ipjM} \right] \ , \tag{3.6}$$

where $\boldsymbol{\mu}_{ipjm}, m = 1, 2, \ldots, M$ is the mean vector of the mth Gaussian component at the jth state of triphone p of base phone i. Similarly, a state supervector $\mathbf{m}_{ij}$ is created from the corresponding monophone state model.

STEP 5 : Let N_i be the number of triphones of base phone i. Collect all triphone state supervectors $\mathbf{v}_{ij1}$, $\mathbf{v}_{ij2}$, ..., $\mathbf{v}_{ijN_i}$ as well as the monophone state supervector $\mathbf{m}_{ij}$ of base phone i, and derive an eigenbasis from their correlation or covariance matrix using *principal component analysis* (PCA). The covariance matrix is computed as follows:

$$\frac{1}{N_i} \sum_{p} (\mathbf{v}_{ijp} - \mathbf{m}_{ij})(\mathbf{v}_{ijp} - \mathbf{m}_{ij})' \ . \tag{3.7}$$

STEP 6 : Arrange the eigenvectors $\{\mathbf{e}_{ijk}, k = 1, 2, \ldots, N_i\}$ in descending order of their eigenvalues λ_{ijk}, and pick the top K_{ij} (where $K_{ij} \leq N_i$) eigenvectors to represent the eigenspace of state j of base phone i.

STEP 7 : Now the supervector $\mathbf{v}_{ijp}$ of state j of any triphone p of base phone i is assumed to lie in the space spanned by the K_{ij} eigentriphones. Thus, we have

$$\mathbf{v}_{ijp} = \mathbf{m}_{ij} + \sum_{k=1}^{K_{ij}} w_{ijpk}\mathbf{e}_{ijk} \ , \tag{3.8}$$

where $\mathbf{w}_{ijp} = [w_{ijp1}, w_{ijp2}, \ldots, w_{ipK_{ij}}]$ is the eigentriphone coefficient vector of state j of triphone p of base phone i.

STEP 8 : Estimate the eigentriphone coefficient vector $\mathbf{w}_{ijp}$ of state j of any triphone p by MLED. Finally, the Gaussian mean of the mth mixture at the jth state of triphone p can be obtained from $\mathbf{v}_{ijp}$ as

$$\boldsymbol{\mu}_{ipjm} = \mathbf{m}_{ijm} + \sum_{k=1}^{K_{ij}} w_{ijpk}\mathbf{e}_{ijkm} \ . \tag{3.9}$$

STEP 9 : If either the eigentriphone coefficients converge or the recognition accuracy of a development data set is maximized, go to STEP 10. Otherwise, re-align the training data using the model in STEP 8, re-estimate the Gaussian means and repeat STEP 4 – 9.

STEP 10 : After the eigentriphone "adaptation" of the Gaussian means, the Gaussian covariances and mixture weights of a triphone are re-estimated if its sample count is greater than the thresholds θ_v and θ_w respectively. Otherwise, they remain the same as those of the monophone state model from which they are cloned.

3.2.3 Cluster-based Eigentriphone

Both the model-based and state-based eigentriphone acoustic modeling methods discussed above derive eigenbases from all triphones of a base phone. In fact, the eigentriphone modeling framework is very flexible and can be applied to any group of phonetic or sub-phonetic units provided that they may be represented by supervectors of the same dimension. For example, if training data are really scarce, one may perhaps derive eigentriphones from broad phonetic classes (such as vowels, fricatives, etc.); on the other hand, when there are sufficient data, one may divide the triphones of a base phone into groups and derive eigentriphones from each triphone group. Thus,

we investigate a more general framework of deriving eigentriphones from clusters of triphones or triphone states , which we call *cluster-based eigentriphone* acoustic modeling. In particular, we investigate eigentriphone modeling with general state clusters.

Common clustering algorithms such as k-means clustering, agglomerative hierarchical clustering, and decision tree, together with a well-defined distance metric or impurity function may be used to generate triphone or state clusters for cluster-based eigentriphone modeling. Instead of delving into various clustering algorithms, we resort to the use of phonetic decision tree for the purpose since it has been applied successfully to a few tasks such as state tying in ASR. In fact, we propose to use the triphone state clusters represented by the nodes in the same state-tying tree for deriving eigentriphones. There are several benefits for the choice:

- In a typical ASR system, there are 39 base phones and the triphone models are 3-state HMMs. Thus, there will be 39 sets of model-based eigentriphones and $39 \times 3 = 117$ sets of state-based eigentriphones. On the other hand, there are many more tied states — usually hundreds or even thousands — in an ASR system, which means that the use of the state clusters from tied-states will allow a higher resolution of eigentriphone modeling than model-based or state-based eigentriphone modeling. Moreover, the state-tying tree gives one the flexibility to decide the modeling resolution by going up or down the phonetic decision tree and choose the right nodes for cluster-based eigentriphone derivation[2].

- State-based eigentriphone modeling is a special case of cluster-based eigentriphone modeling in which each cluster consists of respective states from all triphones of a base phone. However, cluster-based eigentriphone modeling using tied-state clusters is computationally more efficient because the number of tied states is usually much greater than the number of monophone states so that there are fewer triphone state supervectors in each tied-state cluster to derive eigentriphones.

- Most importantly, unseen triphones may also be synthesized using the same phonetic state-tying tree that defines state clusters for cluster-based eigentriphone

[2]Note that the nodes selected for conventional state tying need not be the same as the nodes selected for cluster-based eigentriphone modeling; the two processes simply use the same phonetic decision tree.

modeling as in conventional tied-state triphone HMM systems. That is, since eigentriphone modeling starts with a well-trained tied-state HMM system, the latter will be kept to synthesize unseen triphones as in general practice[3].

- Cluster-based eigentriphone modeling does not require any modification in the tied-state GMM-HMM training procedures. Thus, our method can be viewed as a kind of post-processing that can easily fit into most existing ASR systems.

The basic procedure of cluster-based eigentriphone modeling is described as below. First, we apply decision-tree state clustering to construct a set of conventional tied-state triphone HMMs where each state is represented by an M-component GMM. Then the following procedure is repeated for each state cluster i, consisting of N_i members:

STEP 1 : Untie the Gaussian means of all the triphone states in a state cluster. The means of the cluster GMM are then cloned to initialize *all* untied triphone states in the cluster. Note that the Gaussian covariances and mixture weights of the states in the cluster are still tied together.

STEP 2 : Re-estimate only the Gaussian means of triphone states after cloning. Their Gaussian covariances and mixture weights remain unchanged as their state cluster GMM does.

STEP 3 : Create a triphone state supervector $\mathbf{v}_{ip}$ for each triphone state p in state cluster i by stacking up all its Gaussian mean vectors from its M-component GMM as below

$$\mathbf{v}_{ip} = \left[\boldsymbol{\mu}_{ip1},\, \boldsymbol{\mu}_{ip2},\, \cdots,\, \boldsymbol{\mu}_{ipM} \right] , \tag{3.10}$$

where $\boldsymbol{\mu}_{ipm},\ m = 1, 2, \ldots, M$ is the mean vector of the mth Gaussian component. Similarly, a state cluster supervector $\mathbf{m}_i$ is created from the GMM of state cluster i.

STEP 4 : Collect the triphone state supervectors $\{\mathbf{v}_{i1}, \mathbf{v}_{i2}, \ldots, \mathbf{v}_{iN_i}\}$ as well as the state cluster supervector $\mathbf{m}_i$ of cluster i, and derive an eigenbasis from their covariance or correlation matrix using PCA.

[3]Although eigentriphone modeling opens up another possibility of approximating an unseen triphone by one of the ensuing distinct triphone models that is believed to be most "similar" to the unseen one instead of by the tied states, simple ways to define such similarity did not give better results; further investigation is needed. For simplicity, we still synthesize the unseen triphones using the tied states from the phonetic state-tying tree.

STEP 5 : Arrange the eigenvectors $\{\hat{\mathbf{e}}_{ik}, k = 1, 2, \ldots, N_i\}$ in descending order of their eigenvalues λ_{ik}, and pick the top K_i (where $K_i \leq N_i$) eigenvectors to represent the eigenspace of state cluster i. Note that different state clusters may have different K_i.

STEP 6 : The supervector $\mathbf{v}_{ip}$ of any triphone state p is assumed to lie in the space spanned by the K_i eigentriphones. Thus, we have

$$\mathbf{v}_{ip} = \mathbf{m}_i + \sum_{k=1}^{K_i} w_{ipk}\mathbf{e}_{ik} \ , \tag{3.11}$$

where $\mathbf{w}_{ip} = [w_{ip1}, w_{ip2}, \ldots, w_{ipK_i}]$ is the eigentriphone coefficient vector of triphone p in the "triphone state space" of cluster i.

STEP 7 : Estimate the eigentriphone coefficient vector $\mathbf{w}_{ip}$ of any triphone state p by MLED.

3.3 Extensions to the Basic Procedure

The basic procedure of eigentriphone modeling is motivated from the eigenvoice adaptation framework. However, the eigenvoice adaptation framework is originally developed to solve the speaker adaptation problem. Thus, there are rooms for improvement in the basic procedure of our modeling framework. To make the eigentriphone modeling framework more robust, the modeling framework are improved in the following two aspects:

- In the basic procedure, the triphones are considered "equal" when they are used to derive the eigentriphones. However, due to the uneven distribution of triphones in the training data, some triphones are better trained than others and they are more reliable. Thus, it is desirable to incorporate some notion of triphone reliability in the construction of the eigentriphones.

- After the derivation of eigentriphones, a hard decision is made on the dimensionality of the eigenspace (or, equivalently, the number of eigentriphones), K_i, to represent all the triphone models of base phone i. However, the rationale is that

a triphone with more adaptation data should use a larger K. Thus, it is desirable to deduce an automatic way of finding the K for each adapted triphone.

We propose using weighted PCA [50] and regularization [49] to solve the two problems respectively.

3.3.1 Derivation Using Weighted PCA

The use of *weighted PCA* instead of standard PCA has at least two advantages.

Firstly, it avoids defining and tuning of a sample count threshold to classify the triphones into frequent and infrequent sets. Instead, the use of weighted PCA allows eigentriphones to be derived by taking *all* triphones into account. This is made possible by incorporating some measure of reliability of each triphone in the construction of the eigenbasis that is related to its training data sufficiency. In this thesis, each triphone supervector is weighted by its sample count in the weighted PCA procedure[4]. Thus, for the case of model-based eigentriphone modeling, the covariance matrix in the basic procedure is replaced by

$$\frac{1}{n_i}\sum_p n_{ip}(\mathbf{v}_{ip}-\mathbf{m}_i)(\mathbf{v}_{ip}-\mathbf{m}_i)', \tag{3.12}$$

where n_{ip} is the sample count of the triphone p of base phone i, and $n_i = \sum_p n_{ip}$.

Secondly, as we can see from Fig. 3.2, the eigenvalue spectrum produced by weighted PCA rises more sharply than the spectrum given by standard PCA. It means that fewer leading eigentriphones produced by weighted PCA can capture more variations in the triphone supervectors. As a result, weighted PCA allows the use of fewer eigentriphones in eigentriphone acoustic modeling. This has implications on the space requirement to store the models produced by eigentriphone modeling. Since the models produced by eigentriphone modeling are distinct, each observed triphone — even those with few samples — in the database will be represented by a distinct HMM. Consequently, the model size resulting from eigentriphone modeling is much bigger than conventional tied-state HMMs. With the use of weighted PCA, fewer eigentriphones may be employed to model each triphone, and the model size can be reduced.

[4]In general, the weights may be a function of the sample count n_{ip} such that the weights increase with n_{ip}.

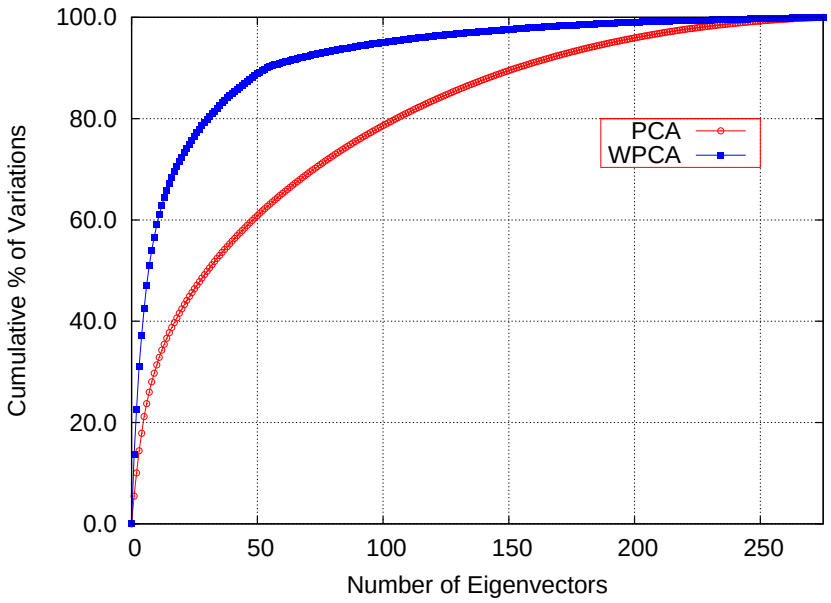

Figure 3.2: Variation coverage by the number of eigentriphones derived from base phone [aa]. The graph is plotted using the WSJ training corpus.

3.3.2 Soft Decision on the Number of Eigentriphones Using Regularization

To avoid making a hard decision on the number of eigentriphones K_i to use for each base phone i, a new penalized log-likelihood function $Q(\mathbf{w}_{ip})$ is defined for the estimation of the interpolation coefficients using *all* eigentriphones:

$$Q(\mathbf{w}_{ip}) = L(\mathbf{w}_{ip}) - \beta R(\mathbf{w}_{ip}) \ , \tag{3.13}$$

where β is the regularization parameter that controls the relative importance of the regularizer $R(\cdot)$ compared with the likelihood term $L(\cdot)$ of Eqn. (3.4). The regularizer should be chosen so that the more informative eigentriphones (with larger eigenvalues) are automatically emphasized and the less informative eigentriphones (with smaller eigenvalues) are automatically de-emphasized. In this thesis, the following regularizer is found to be effective

$$R(\mathbf{w}_{ip}) = \sum_{k=1}^{N_i} \frac{w_{ipk}^2}{\lambda_{ik}} \ . \tag{3.14}$$

The proposed regularizer represents a scaled Euclidean distance of the triphone from the base phone in the space spanned by the eigentriphones. It has the following properties:

- The squared coefficient of each eigentriphone, w_{ipk}, is inversely scaled by its eigenvalue so that a less informative eigentriphone will have less influence on the "adapted" triphone model.

- When there are a lot of training data, the likelihood term will dominate the objective function $Q(\mathbf{w}_{ip})$, and the "adapted" triphone model will converge to its conventional Baum-Welch training estimate.

- On the other hand, for a triphone with a limited amount of training data, the penalty term will dominate and a smaller scaled Euclidean distance between the triphone and base phone is preferred. In other words, its "adapted" triphone model will fallback to its monophone model.

Thus, in effect, the regularizer of Eqn. (3.14) will provide a soft decision on the number of eigentriphones to use for each triphone (and not just for each base phone).

Differentiating the optimization function $Q(\mathbf{w}_{ip})$ of Eqn. (3.13) w.r.t. each eigentriphone coefficient w_{ipk}, and setting each derivative to zero, we have,

$$\sum_{n=1}^{N_i} A_{ipkn} w_{ipn} + \beta \frac{w_{ipk}}{\lambda_{ik}} = B_{ipk} \quad \forall k = 1, 2, \cdots N_i \tag{3.15}$$

where

$$\begin{aligned} A_{ipkn} &= \sum_{j,m} \mathbf{e}'_{ikjm} C^{-1}_{ipjm} \mathbf{e}_{injm} \left(\sum_t \gamma_{ipjm}(t) \right) \\ B_{ipk} &= \sum_{j,m} \mathbf{e}'_{ikjm} C^{-1}_{ipjm} \left(\sum_t \gamma_{ipjm}(t) (\mathbf{x}_t - \mathbf{m}_{ijm}) \right) . \end{aligned}$$

The eigentriphone coefficients may easily be found by solving the system of N_i linear equations represented by Eqn. (3.15), and the Gaussian means of the new model may be computed using Eqn. (3.9).

As pointed out in Section 3.3.1, weighted PCA may allow pruning of eigentriphones in the final models. When that is performed, the proposed soft decision on the number of eigentriphones using regularization is applied on *all* the remaining eigentriphones *after* eigentriphone pruning. We called this new way of estimating coefficients *penalized maximum-likelihood eigen-decomposition* (PMLED).

3.4 Experimental Evaluation

Our proposed eigentriphone modeling method was evaluated on two speech recognition tasks: phoneme recognition on TIMIT [99] and medium-vocabulary continuous speech recognition on the Wall Street Journal (WSJ) [75] 5K task.

In both tasks, we compare the performance of the following five acoustic modeling methods:

- baseline1: conventional Baum-Welch training of triphone HMMs with no state tying.
- baseline2: conventional Baum-Welch training of tied-state triphone HMMs.
- model-based eigentriphone modeling of triphone HMMs as described in Section 3.2.1 (with no tied states).
- state-based eigentriphone modeling of triphone HMMs as described in Section 3.2.2 (with no tied states).
- cluster-based eigentriphone modeling of triphone HMMs using tied-state clusters as described in Section 3.2.3 (but no tied states).

Cross-word triphones[5] were employed in all experiments and modeled as continuous-density hidden Markov models (CDHMMs). Each CDHMM was a 3-state strictly left-to-right HMM in which the state distributions were modeled by a mixture of 16 Gaussians with diagonal covariances. In addition, there were a 1-state short pause model and a 3-state silence model whose middle state was tied to the short pause state. Feature vectors were standard 39-dimensional MFCC acoustic vectors, and they were extracted from the training speech data every 10ms over a window of 25ms. The HTK toolkit [98] was used for HMM training and decoding with a beam width of 350.

All eigentriphone modeling experiments employed (weighted) PCA using correlation matrices, and the soft decision on the number of eigentriphones with the use of regularization to determine the eigentriphone coefficients. All system parameters such

[5]Triphones are constructed across word boundaries.

as the regularization parameter β, grammar factor, insertion penalty, as well as the optimal number of tied states for conventional HMM training, and the optimal number of state clusters for cluster-based eigentriphone modeling were determined using the respective development data set[6].

Table 3.1: Information of TIMIT data sets.

Data Set	#Speakers	#Utterances	#Hours
training	462	3,696	3.14
core test	24	192	0.16
development	24	192	0.16

3.4.1 Phoneme Recognition on TIMIT

3.4.1.1 Speech Corpus and Experimental Setup

The standard TIMIT training set which consists of 3,696 utterances from 462 speakers was used to train the various models, whereas the standard core test set which consists of 192 utterances spoken by 24 speakers was used for evaluation. The development set is part of the complete test set, consisting of 192 utterances spoken by 24 speakers. Speakers in the training, development, and test set do not overlap. A summary of these data sets is shown in Table 3.1.

We followed the standard experimentation on TIMIT, and collapsed the original 61 phonetic labels in the corpus into a set of 48 phones for acoustic modeling; the latter were further collapsed into the standard set of 39 phonemes [61] for error reporting. Moreover, the glottal stop [q] was ignored. At the end, there were altogether 15,546 cross-word triphone HMMs based on 48 base phones. Phoneme recognition was performed using Viterbi decoding with a trigram phone language model (LM) that was trained from the TIMIT training transcriptions using the SRILM language modeling toolkit [87]. The trigram LM has a perplexity of 14.39 on the core test set.

[6]Firstly, the grammar factor and the insertion penalty were optimized in the conventional tied-state HMM system, and then used without modification in other systems. Secondly, for different number of state clusters, cluster-based eigentriphone modeling was run and the regularization parameter β was tuned, all using the development data. Finally, the number of state clusters that gave the best recognition result for the development data was recorded.

3.4.1.2 Acoustic Modeling

Five sets of triphone HMMs were built according to the five acoustic modeling methods mentioned at the beginning of Section 3.4. For the conventional tied-state triphone HMM system (baseline2), there are a total of 587 tied states[7]. The dimension of triphone supervectors in model-based eigentriphone modeling is $3(\text{states}) \times 16(\text{mixtures}) \times 39(\text{MFCC}) = 1,872$. The dimension of triphone supervectors in state-based or cluster-based eigentriphone modeling is $16(\text{mixtures}) \times 39(\text{MFCC}) = 624$. The number of bases for the model-based, state-based and cluster-based eigentriphone modeling is 44, 132 and 587 respectively[8]. In fact, cluster-based eigentriphone modeling was conducted using the clusters defined by the same 587 tied states in the baseline2 system.

3.4.1.3 Results and Discussion

Table 3.2: Phoneme recognition accuracy (%) of various systems on TIMIT core test set using phone-trigram language model.

Model	Accuracy
baseline1: conventional training (no state tying)	68.63
baseline2: conventional tied-state HMM training	71.95
model-based eigentriphone training model (no state tying)	71.27
state-based eigentriphone training model (no state tying)	71.03
cluster-based eigentriphone training model (587 state clusters)	72.90

Phoneme recognition results of the five systems are compared in Table 3.2.

Though states are not tied in the three eigentriphone modeling methods, they outperform conventional HMM training without state tying by 3–4% absolute. In fact, their phoneme recognition performance is comparable to conventional tied-state HMM training, and cluster-based eigentriphone modeling actually outperforms the latter by an absolute of 1%.

[7]The number of tied states was selected to maximize the phoneme recognition accuracy of the development set. It turns out the number is close to but not optimal on the core test set. (See Fig. 3.4.)

[8]Among the 48 phones that were selected for acoustic modeling, four phones are different variants of silence and closure, and they were modeled as monophone HMMs.

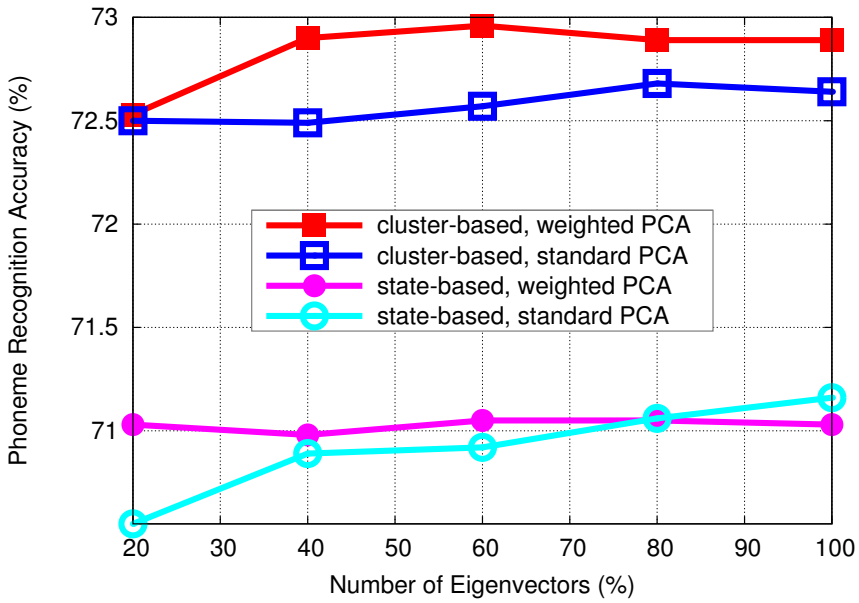

Figure 3.3: Improvement of cluster-based eigentriphone modeling over state-based eigentriphone modeling on TIMIT phoneme recognition.

Among the three eigentriphone modeling methods, the cluster-based method is the best, followed by the model-based method and then the state-based method. Both the model-based method and state-based method estimate eigenbases from all triphones of a base phone, but the former method concatenates the three state supervectors of each triphone into one long triphone supervector for basis derivation. The better performance of the model-based method suggests that better eigenbases may be produced by making use of the correlation among the triphone HMM states. On the other hand, both the state-based method and the current cluster-based method create eigenbases at the state level. The better performance of the cluster-based method must be attributed to the higher modeling resolution — 587 state clusters in the cluster-based method versus 132 state clusters in the state-based method — which more than compensates for the loss of state correlation that is otherwise maintained in the model-based method, and gives the best performance.

We further compared the performance of state-based eigentriphone modeling with cluster-based eigentriphone modeling when different forms of PCA were used, and when different proportions of eigentriphones were pruned. Eigentriphone pruning was done by first arranging the eigentriphones of each basis in descending order of their eigenvalues, and then retaining different numbers of leading eigentriphones for mod-

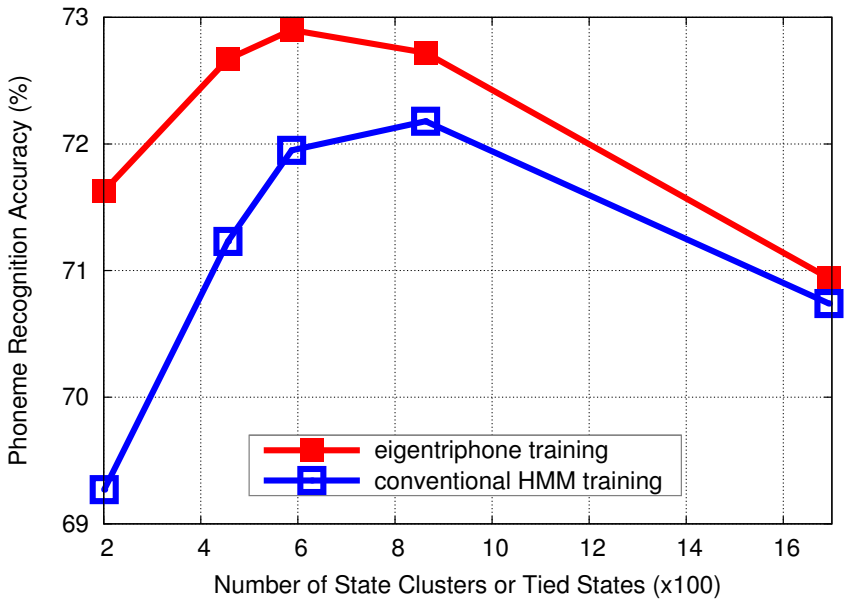

Figure 3.4: TIMIT phoneme recognition performance of cluster-based eigentriphone modeling and conventional tied-state HMM training with varying number of state clusters or tied states.

eling the triphones. The result is shown in Fig. 3.3. Since the cluster-based method employs more state clusters than the state-based method (587 vs. 132), the former creates about four times more eigenbases than the latter. Equivalently, the number of eigentriphones in each eigenbasis produced by the former is only about $1/4$ of the latter on average. However, according to Fig. 3.3, one may still prune 60% of the eigentriphones in both methods without any performance loss[9]. The figure also shows that weighted PCA is more effective than standard PCA in deriving the eigentriphones in both methods.

Table 3.2 only shows the best results of various systems under the optimal settings determined by the development set. The effect of the number of state clusters on cluster-based eigentriphone modeling was also studied and compared with the effect of using different numbers of tied states on conventional HMM training as shown in Fig. 3.4. The results show that for the same number of state clusters (or tied states), cluster-based eigentriphone modeling always performs better than conventional tied-state HMM training, and the optimal number of state clusters is similar for both acoustic modeling methods. The difference between the two curves in the figure represents

[9]Note that 40% of eigentriphones in the cluster-based eigentriphone modeling method contain fewer eigentriphones than 40% of eigentriphones in the state-based eigentriphone modeling method. Specifically, the former is about $1/4$ of the latter.

the amount of quantization error that is recovered by the current cluster-based eigentriphone modeling method. The result of a significant test run on the performance difference between conventional tied-state HMM training and cluster-based eigentriphone training is shown in Table B.1 in Appendix B.

Table 3.3: Information of WSJ data sets. The out-of-vocabulary (OOV) is computed with respect to to the 5K vocabulary defined in the recognition task.

Data Set	#Speakers	#Utterances	Vocab Size	OOV	LM Perplexity
SI284	283	37,413	13,646	11.95%	—
si_dt_05.odd	10	248	1,260	0	—
Nov'92	8	330	1,270	0	56.94
Nov'93	10	215	1,004	0.29%	61.82

3.4.2 Word Recognition on Wall Street Journal

3.4.2.1 Speech Corpus and Experimental Setup

The standard SI-284 Wall Street Journal (WSJ) training set was used for training the speaker-independent model. It consists of 7,138 WSJ0 utterances from 83 WSJ0 speakers and 30,275 WSJ1 utterances from 200 WSJ1 speakers. Thus, there is a total of about 70 hours of read speech in 37,413 training utterances from 283 speakers. All the training data are endpointed. The standard Nov'92 and Nov'93 5K non-verbalized test set were used for evaluation using the standard 5K-vocabulary trigram language model (LM) that came with the WSJ corpus. The set si_dt_05.odd contains alternate sentences from the 1993 WSJ 5K Hub development test set after sentences with OOV words were removed. This was used to tune the system parameters. A summary of these data sets is shown in Table 3.3.

3.4.2.2 Acoustic Modeling

There were 18,777 cross-word triphones based on 39 base phones. For the conventional tied-state system (baseline2), the best performance was obtained with 7,374 tied states. The dimension of triphone supervectors in model-based, state-based, and cluster-based eigentriphone modeling are the same as those in the TIMIT experiment,

Table 3.4: Word recognition accuracy (%) of various systems on the WSJ 5K task using trigram language model.

Model	Nov'92	Nov'93
baseline1: conventional training; no state tying	95.61	94.05
baseline2: conventional tied-state HMM training	96.32	94.21
model-based eigentriphone training model	96.26	94.52
state-based eigentriphone training model	95.87	94.15
cluster-based eigentriphone training model	96.32	94.54

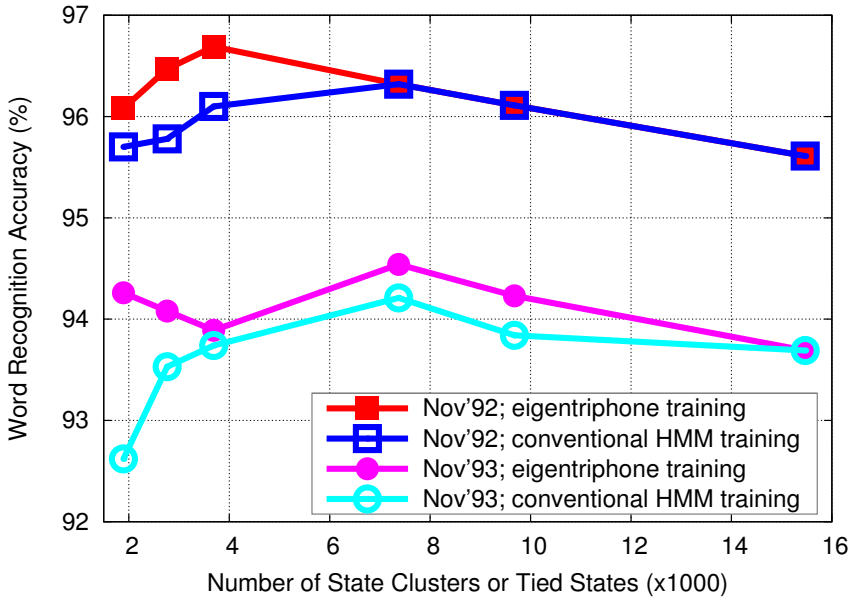

Figure 3.5: WSJ recognition performance of cluster-based eigentriphone modeling and conventional tied-state HMM training with varying number of state clusters or tied states.

namely 1872, 624, and 624, respectively; the number of bases for the three methods is 39, 117, and 7,374 respectively.

3.4.2.3 Results and Discussion

The word recognition results of various systems are shown in Table 3.4.

Comparing the performance of baseline1 and baseline2, we once again observe the effectiveness of state tying in triphone acoustic modeling. However, eigentriphone modeling can be an alternative: all the three variants of the method give comparable, if not better, recognition performance on WSJ. The state-based method is again the weakest among the three eigentriphone modeling methods; the model-based

method and the cluster-based method have similar performance with the latter being slightly better. On the Nov'92 test set, the cluster-based eigentriphone modeling method has the same word recognition accuracy as the conventional tied-state HMM training method, but on the Nov'93 test set, the former actually reduces the word error rate of the latter by 5.7%.

The performance of the cluster-based eigentriphone modeling method and conventional tied-state HMM training method over varying number of state clusters or tied states was also studied. The results are shown in Fig. 3.5. It can be seen that cluster-based eigentriphone modeling always performs better than conventional tied-state HMM training for the same number of state clusters or tied states. Note that on the Nov'92 test set, cluster-based eigentriphone modeling may achieve a better result of 96.69% word accuracy by using 3,690 state clusters. The worse result of the method in Table 3.4 was obtained with 7,374 state clusters which were found to be optimal on the development set. The results of significant tests run on the performance difference between conventional tied-state HMM training and cluster-based eigentriphone training are shown in Table B.2 and Table B.3 in Appendix B.

Table 3.5: Count of infrequent triphones in the test sets of TIMIT and WSJ for different definition of infrequency. The WSJ figures here refer to SI284 training set.

Sample Count Below	Nov'92	Nov'93	TIMIT
10	0.82%	0.94%	26.29%
20	1.75%	2.13%	40.66%
30	2.55%	3.01%	50.36%
40	3.55%	3.99%	57.10%
50	4.53%	5.15%	61.89%

3.4.3 Analysis

We further analyze the different behaviour of eigentriphone modeling in the two tasks investigated in this proposal.

3.4.3.1 Different Recognition Improvements in TIMIT and WSJ

From the recognition results of TIMIT (Table 3.2) and WSJ (Table 3.4), it is noticed that the performance gain is much more noticeable in TIMIT than in WSJ, and the gain

in Nov'93 is also higher than in Nov'92.

Since the utmost strength of our new eigentriphone acoustic modeling method is its ability to construct distinct models for each seen triphone — both frequent and infrequent triphones — in the training set robustly, we believe that the performance gain in a task should depend on how often those triphones that are infrequent in the training set appear in the corresponding test set. If more of these infrequent triphones appear in the test set, and if they are better estimated by eigentriphone modeling than by conventional tied-state HMM training, then the performance gain by eigentriphone modeling should be higher. Thus, we count the amount of infrequent triphones in the test sets of TIMIT and WSJ with different definitions of infrequency, and the finding is summarized in Table 3.5.

From the table, it can be seen that infrequent triphones appear much more in TIMIT than in WSJ [10]. For example, if we consider triphones that appear fewer than 30 times in the training set as infrequent triphones, which are more likely to be under-trained, then 50.36% of the triphones in the TIMIT test set are infrequent in the TIMIT training set, whereas the corresponding figures for WSJ Nov'92 and Nov'93 are 2.55% and 3.01% respectively. In the baseline tied-state HMM system, these infrequent triphones are modeled by tied-state HMMs, while in our new eigentriphone systems, they are distinctly modeled as linear combinations of eigentriphones. Since the infrequent triphones occur rarely in WSJ test sets, the advantage of eigentriphone modeling is small. On the other hand, infrequent triphones appear much more in the test set of TIMIT, thus the performance gain by eigentriphone modeling over conventional tied-state HMM training is more obvious and significant in TIMIT.

3.4.3.2 Effectiveness of Adapting Infrequent Triphones

In order to show the effectiveness of adapting infrequent triphones with our proposed method, we would like to repeat the WSJ experiment with a smaller training set. Here the SI84 WSJ training set is used. It consists of the 7,138 WSJ0 utterances from only 83 WSJ0 speakers and is thus a subset of the SI284 training set. There are about 14 hours

[10] This is because the purposes of the two tasks are different. TIMIT dataset is collected for phoneme recognition and it aims to cover a larger diversity of phoneme combinations. In contrast, WSJ is designed for continuous speech recognition and the sentences chosen in the task have normally limited the percentage of infrequent triphones.

Table 3.6: Word recognition accuracy (%) on the WSJ Nov'92 5K task using the SI84 training set and a bigram language model. θ_m = 30 means only triphones with more than 30 samples will be adapted. The remaining triphones were copied from the conventional tied-state system.

Model	θ_m	Nov'92
baseline2: conventional tied-state HMM training	-	93.09
cluster-based eigentriphone training model	50	93.29
	40	93.33
	30	93.33
	20	93.42
	10	93.67
	0	93.89

Table 3.7: Performance of cluster-based eigentriphone modeling and conventional tied-state triphones using different WSJ training sets. Recognition has done on the WSJ Nov'92 5K evaluation set using a bigram language model.

Training Set	Tied-state	Cluster-based ETM
SI84	93.09	93.89
SI284	94.25	94.30

Table 3.8: Count of infrequent triphones in the WSJ nov'92 test set with respect to different training set.

Sample Count Below	Nov'92(w.r.t. SI84)	Nov'92(w.r.t. SI284)
10	5.37%	0.82%
20	11.17%	1.75%
30	15.67%	2.55%
40	19.47%	3.55%
50	23.49%	4.53%

of read speech in this training set. The accuracy results of cluster-based eigentriphone modeling with different values of θ_m are shown in Table 3.6. θ_m is the threshold where only triphones with number of samples more than θ_m will be updated. From the results, we can see that the error rate reduction becomes smaller when triphones with the fewest samples are not adapted. Thus, we believe that most of the gain with eigentriphone modeling is attributed to the better estimation of the infrequent triphones.

Table 3.7 compares the performance of cluster-based eigentriphone modeling with different training sets. First of all, it is not surprising to see that the overall accuracy of using the SI84 training set is lower than using SI284 as the SI84 training set is only about 1/5 of the SI284 training set. However, our method obtained a higher error rate reduction when the SI84 training set was used. This can be explained by the higher poor triphone rate in Table 3.8 when a smaller training set is used.

3.4.3.3 Effectiveness of Using PMLED

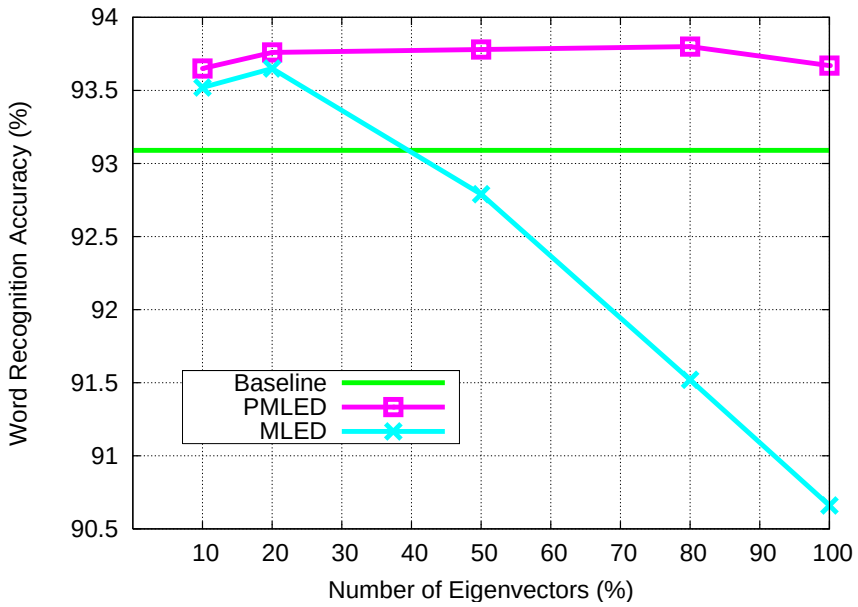

Figure 3.6: Comparison between PMLED and MLED when different proportions of eigentriphones are used.

The WSJ recognition task with the use of the SI84 training set is again repeated to show the effectiveness of using PMLED to determine the eigentriphone coefficients. The comparison of using PMLED and the classical MLED on cluster-based eigentriphone modeling with different proportions of eigentriphones is shown in Fig. 3.6.

From the figure, we observe the following:

- The performance of classical MLED is sensitive to the number of eigentriphones. The accuracy falls rapidly when more than 20% of eigentriphones are used. It is because when there is too much eigentriphones, the triphones with small amount of training data cannot be robustly estimated. When all the eigentriphones are used, the triphones included in the reference set will fall back to their conventional ML estimates.
- When PMLED is used, the accuracy is less sensitive to the number of eigentriphones as the regularization avoids making a hard decision on the number of eigentriphones. The overall performance of using PMLED is better than the system that uses the best number of eigentriphones.

Table 3.9: Computational requirements during decoding by the models estimated by conventional HMM training and cluster-based eigentriphone modeling. (See text for details)

ASR Task	Comparison	Conventional HMM Training	Eigentriphone Modeling
TIMIT	#Distinct States	587	46,638
	Memory (MB)	1.47	49.2
	Relative Decode Time	1.00	1.80
WSJ	#Distinct States	7,374	56,331
	Memory (MB)	18.4	75.3
	Relative Decode Time	1.00	1.25

3.4.3.4 Additional Computational Requirements

There is a price to pay for the better performance of eigentriphone modeling. Since the triphone models produced by eigentriphone modeling are all distinct, their model size is much bigger than the models produced by conventional tied-state HMM training. Throughout the decoding process, the triphone state space of eigentriphone-constructed models is also much larger. Table 3.9 compares the number of distinct triphone states, memory used to store Gaussian mean vectors[11] assuming 60% eigentriphone pruning, and the decoding time of eigentriphone-constructed models relative

[11]The memory requirement of transition probabilities and Gaussian variances are not considered here as cluster-based eigentriphone modeling copies them from the conventional tied-state HMMs. Hence, the memory requirements of these quantities for both training methods are the same.

to that of conventional tied-state models. Although the model size and state space of eigentriphone-constructed models are much larger than those of conventional HMMs, the increase in decoding time is disproportionally small. This is due to the disproportionally small increase in the number of active states during decoding. In addition, the increase in decoding time is larger in TIMIT than in WSJ because of the much larger number of triphones in the TIMIT phone decoding network.

3.5 Evaluation with Discriminatively Trained Baseline

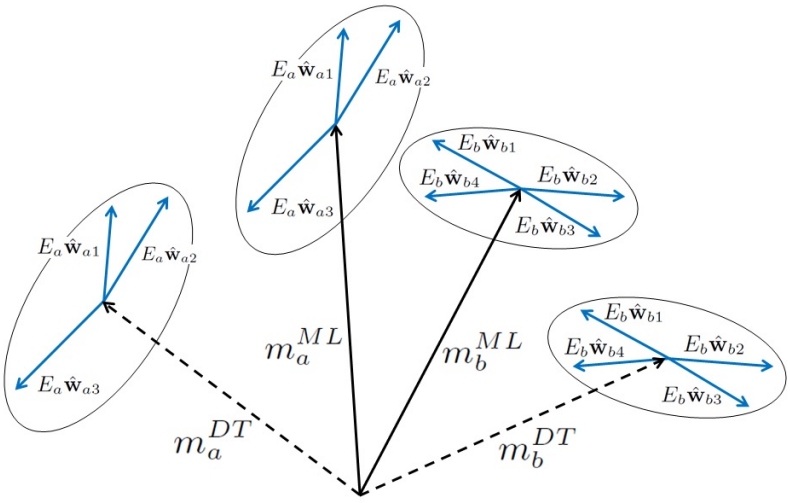

Figure 3.7: An illustration of the inter-cluster and intra-cluster discriminations provided by discriminative training and cluster-based eigentriphone modeling respectively. m_a^{ML} and m_b^{ML} are the centers of cluster a and b obtained through ML training; m_a^{DT} and m_b^{DT} are the centers of cluster a and b obtained through discriminative training.

As shown in Fig. 3.7, the discrimination among triphone states within the same state cluster is now modeled by an addition of vectors $\mathbf{E}_i\hat{\mathbf{w}}_{ip}$ through our cluster-based eigentriphone modeling:

$$\mathbf{v}_{ip} = \mathbf{m}_i^{ML} + \mathbf{E}_i\hat{\mathbf{w}}_{ip} \ , \tag{3.16}$$

where $\mathbf{E}_i = [\mathbf{e}_{i1}, \ldots, \mathbf{e}_{iK_i}]$ is the matrix of eigentriphones that is used to model the intra-cluster discrimination among the member states cluster i, and $\hat{\mathbf{w}}_{ip} = [w_{ip1}, \ldots, w_{ipK_i}]'$ is the eigentriphone coefficient vector of triphone state p in the cluster. Meanwhile, the discrimination among state clusters is modeled by the cluster mean vector $\mathbf{m}_i^{ML}$ which

are obtained through ML training. Nevertheless, the inter-cluster discrimination can be readily enhanced by conventional discriminative training (DT). Thus, we would like to investigate the combination of two approaches by replacing $\mathbf{m}_i^{ML}$ with a set of discriminative trained biases $\mathbf{m}_i^{DT}$. This could be achieved by directly bootstrapping our cluster-based eigentriphone modeling from a discriminatively trained GMM-HMM.

3.5.1 Experimental Setup

The performance (in term of word accuracy) of the following four acoustic modeling methods are compared on the WSJ recognition tasks:

- baseline1: conventional ML training of tied-state triphone HMMs.
- baseline2: minimum-phone-error (MPE) discriminative training of tied-state triphone HMMs resulted from baseline1.
- cluster-based eigentriphone modeling of triphone HMMs bootstrapping from baseline1.
- cluster-based eigentriphone modeling of triphone HMMs bootstrapping from baseline2.

All the above systems are trained by the WSJ SI84 training set. Both baseline1 and baseline2 consist of 1,277 tied-states. The cluster-based eigentriphone modeling was conducted using the clusters defined by the tied states in the baseline systems with implementation of PMLED and WPCA. Trigram language model is used for recognition.

3.5.2 Results and Discussion

Word recognition results of various systems are compared in Table 3.10. First of all, we can see that classical discriminative training (baseline2) obtains a greater improvement than our cluster-based eigentriphone modeling over conventional ML training of tied-state triphones (baseline1). This suggests that exploiting inter-discrimination of tied states might be more significant than achieving intra-discrimination among the members of a tied state. Nevertheless, bootstrapping cluster-based eigentriphone modeling from discriminatively trained triphones (baseline2) obtain a further improvement

Table 3.10: Recognition word accuracy (%) of various systems trained by SI84 training set on the WSJ Nov'92 5K evaluation set using trigram language model.

Model Description	Accuracy
Baseline1: tied-state triphones (ML)	95.46
Baseline2: tied-state triphones (MPE)	95.78
Cluster-based eigentriphone modeling on baseline1	95.68
Cluster-based eigentriphone modeling on baseline2	96.06

of 0.28% absolute and a word error rate reduction of 6.6%. This suggests that the gains from the two approaches are supplementary to each other.

CHAPTER 4

EIGENTRIGRAPHEMES FOR SPEECH RECOGNITION OF UNDER-RESOURCED LANGUAGES

In Chapter 3, we investigated the use of distinct acoustic modeling in a phone-based system and hence our proposed method is called *eigentriphone modeling*. One major advantage of our proposed method is that the framework is very flexible and can be applied to any group of modeling unit provided that they may be represented by vectors of the same dimension. Thus, we would like to test the flexibility of our proposed distinct acoustic modeling framework. On the other hand, although phone-based modeling is the mainstream in ASR, grapheme-based modeling is useful in ASR for under-resourced languages of which the phonetics and linguistics are not well studied. Similar to phone-based modeling, parameter tying is also widely used in grapheme-based modeling. It is therefore worth investigating the use of distinct modeling in a grapheme-based system.

In this chapter, we first give an introduction to under-resourced languages and the challenges in their recognition. Then we describe the conventional grapheme-based modeling and our proposed method named *cluster-based eigentrigrapheme modeling* which is based on the same methodology of *cluster-based eigentriphone modeling*. Four under-resourced official South African languages (Afrikaans, South African English, Sesotho, siSwati) were used in a series of speech recognition experiments to demonstrate the effectiveness of *eigentrigrapheme modeling* over conventional acoustic modeling methods.

4.1 Introduction to Automatic Speech Recognition of Under-Resourced Languages

In the past, research efforts on automatic speech recognition (ASR) have been highly focused on the most popular languages such as English, Mandarin, Japanese, French,

German, and so on, in the developed countries. The remaining world languages, lacking audio and language resources, are considered under-resourced languages. Usually the phonetics and linguistics of these languages are not well studied either, thus the development of human language technologies for these languages have been greatly hindered. Nonetheless, some of these under-resourced languages are spoken by large populations. For example, Vietnamese is spoken by about 80 million people, and Thai is spoken by 60 million people. It is not difficult to see that real-life ASR applications for these languages have great potential. One major obstacle in developing an ASR system for under-resourced languages is the availability of data. It is usually costly and labor-intensive to create audio recordings and their human annotated transcriptions, and make linguistic analyses for languages. As a consequence, it is both academically intriguing and commercially attractive to look for more economically efficient and faster ways to create human language technologies for under-resourced languages.

In order to reduce the amount of annotated audio data for training the acoustic models of a new target language, cross-lingual [72, 58] and multi-lingual [53] acoustic modeling techniques have been developed. The rationale behind these techniques is that an acoustic model may be ported to or adapted from some other high-resourced languages, and only a relatively small amount of training data is required for the target language. A key step for these cross-lingual or multi-lingual techniques to work is to figure out a good mapping between phonemes across the languages. This can be done using either a knowledge-based [26] or a data-driven approach [53]. In the data-driven approach, the similarities between sounds can be measured by various distance measures such as confusion matrix [26], entropy-based distance [53] or Euclidean distance [86]. The approach is further improved when the underlying model is more compactly represented. A notable example is the use of subspace Gaussian mixture model (SGMM) [24] in multi-lingual ASR [23, 64]. Another research direction is heading toward making linguistic analysis of a target language easier and faster. Deducing the phone set and preparing the pronunciation dictionary for a new language usually require native linguistic experts. This process is expensive and time-consuming, and is even more so for non-native developers. One way to partially automate the development of a pronunciation dictionary is to first prepare a small primary dictionary manually, and then use it to bootstrap a large dictionary by applying grapheme-to-phoneme

conversion [66, 7, 17, 3]. However, the performance of the final dictionary depends highly on the quality of the primary one. If the primary dictionary is not rich enough and does not cover all the implicit grapheme-to-phoneme relations in the language, the performance of the overall system will be badly influenced.

On the other hand, there is a simple solution to the creation of the phone set and pronunciation dictionary for an under-resourced language: there is no need to develop them if graphemes instead of phonemes are adopted as the acoustic modeling units. In grapheme modeling [80, 47, 73, 58], each word in the "pronunciation dictionary" is simply represented by its graphemic transcription according to its lexical form. According to [16], there are six types of writing systems in the world: logosyllabary, syllabary, abjad, alphabet, abugida, and featural. Many languages that use the alphabet writing system are suitable for grapheme acoustic modeling, and their grapheme set is usually selected to be the same as their alphabet set [80].

The performance of grapheme modeling in ASR is sensitive to the languages. For example, it works better than phone modeling in Spanish but worse than phone modeling in English and Thai [89]. The reason is that the pronunciation of English has developed away from its written form over time, whereas Thai has some complex rules that map its writing to the pronunciation. There are techniques that improve grapheme modeling; for example, in [73], a text normalization scheme was applied on Thai graphemes to improve the performance of a Thai ASR system. There are also works on multi-lingual grapheme modeling [88, 48]. These techniques, however, are usually language-dependent as linguistic knowledge of the target language has to be known in advance. Thus, it is favourable to investigate a language-independent technique to improve current grapheme modeling.

In conventional grapheme-based acoustic modeling, context-dependent trigraphemes [1] are used as the modeling units. Similar to the case of using triphones, state tying are widely used to cluster the trigraphemes and the members in the same cluster share the parameters. In this chapter, we would like to investigate the framework of distinct acoustic modeling in modeling context-dependent graphemes and we call our new method *cluster-based eigentrigrapheme acoustic modeling*.

[1] Trigraphemes are developed from context-independent graphemes by taking the preceding and following graphemes into consideration.

4.2 Cluster-based Eigentrigrapheme Acoustic Modeling

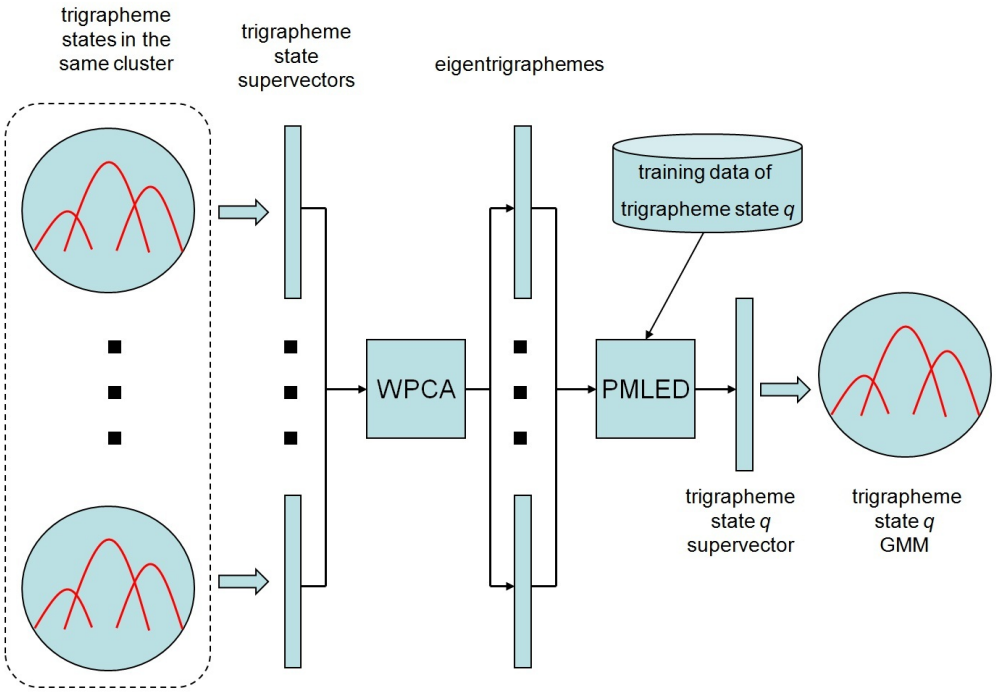

Figure 4.1: The cluster-based eigentrigrapheme acoustic modeling method. (WPCA = weighted principal component analysis; PMLED = penalized maximum-likelihood eigen-decomposition)

Fig. 4.1 shows an overview of the cluster-based eigentrigrapheme acoustic modeling method. The framework is very similar to the cluster-based eigentriphone modeling method described in Chapter 3. All trigrapheme states are first represented by some supervectors and they are assumed to lie in a low dimensional space[2] spanned by a set of eigenvectors. In other words, each trigrapheme supervector is a linear combination of a small set of eigenvectors which are now called eigentrigraphemes. Clustering of the states can be done by a singleton decision tree, and the procedure is exactly the same as that of creating a conventional tied-state trigrapheme system.

Cluster-based eigentrigrapheme modeling consists of three major steps: (a) state clustering via a singleton decision tree, (b) derivation of the eigenbasis, and (c) estimation of eigentrigrapheme coefficients. They are discussed in further detail in the

[2]The dimension of the space is low when compared with the dimension of the trigrapheme state supervectors.

following.

4.2.1 Trigrapheme State Clustering (or Tying) by a Singleton Decision Tree

One major difference between phone-based and grapheme-based acoustic modeling lies in the construction of the decision tree for tying hidden Markov model (HMM) states. In phone-based modeling, it is well-known that decision tree using phonetic questions [79] can significantly improve speech recognition performance by striking a good balance between the trainability and resolution of the acoustic models. However, it is not clear how the phonetic questions used in a phone-based system to tie triphone states can be ported to tie trigrapheme states in a grapheme-based system as the relation between the graphemes and their influence in the pronunciation of their neighboring graphemes is not well understood. In [80], different types of questions are investigated [47] and it is found that questions asking only the identity of the immediate neighboring grapheme, named as *singleton questions*, work at least as well as other types of questions. In this thesis, decision tree[3] using singleton questions at each node is used to generate the conventional tied-state trigrapheme HMMs. In addition, the trigrapheme states that belong to the same tied state naturally form a state cluster on which our new cluster-based eigentrigrapheme modeling may be applied. In other words, the same singleton decision tree can be used to create the tied states for a conventional tied-state trigrapheme system as well as the state clusters for the construction of cluster-based eigentrigraphemes[4].

4.2.2 Conventional Tied-state Trigrapheme HMM Training

We adopt the standard procedure in HTK [98] to create a conventional tied-state trigrapheme HMM system as follows.

[3]The questions in the decision tree are generated from the grapheme set of the target language which is derived by scanning through the training data. Thus, the trees are language-dependent but our method is still language-independent.

[4]Although the state clusters of cluster-based eigentrigrapheme modeling and the tied states of conventional trigrapheme modeling both come from the nodes of the same decision tree, in general, they may not be exactly the same nodes. The optimal set of tied states or state clusters is determined using a separate set of development speech data.

STEP 1 : Context-independent grapheme acoustic models are estimated from the training data. Each context-independent grapheme model is a 3-state strictly left-to-right HMM, and each state is represented by a single Gaussian.

STEP 2 : Each context-independent grapheme HMM is then cloned to initialize *all* its context-dependent trigraphemes.

STEP 3 : For each base grapheme, the transition probabilities of all its trigraphemes are tied together.

STEP 4 : For each base grapheme, tie the corresponding HMM states of all its trigraphemes using a singleton decision tree. Thus, three singleton decision trees are built for each base grapheme. Once a set of trigrapheme states are tied together, they share the same set of Gaussian means, diagonal covariances, and mixture weights.

STEP 5 : Synthesize the unseen trigraphemes by going through the singleton questions of the decision trees.

STEP 6 : Grow the Gaussian mixtures of the models with the training data until each tied state is represented by an M-component Gaussian mixture model (GMM) with diagonal covariance. In practice, the optimal value of M is determined by a separate set of development data.

4.2.3 Eigentrigrapheme Acoustic Modeling

Recall that each node in the state clustering decision tree has a dual role: it is treated as a tied state for tied-state HMM training, and as a state cluster for eigentrigrapheme modeling. To begin cluster-based eigentrigrapheme modeling, one first decides which tree nodes are to be used as the state clusters. Then the state-clusters are treated as tied states, and conventional tied-state trigrapheme HMMs are created using the procedure described in Section 4.2.2. The resulting tied-state HMMs are used as the initial models for deriving the eigentrigraphemes of each state cluster.

4.2.3.1 Derivation of Cluster-based Eigentrigraphemes

The following procedure is repeated for each state cluster i, consisting of N_i member states.

STEP 7 : Untie the Gaussian means of all the trigrapheme states in a state cluster with the exception of the unseen trigrapheme states. The means of the cluster GMM are then cloned to initialize *all* untied trigrapheme states in the cluster. Note that the Gaussian covariances and mixture weights of the states in the cluster are still tied together.

STEP 8 : Re-estimate only the Gaussian means of trigrapheme states after cloning. Their Gaussian covariances and mixture weights remain unchanged as their state cluster GMM does.

STEP 9 : Create a trigrapheme state supervector $\mathbf{v}_{ip}$ for each trigrapheme state p in state cluster i by stacking up all its Gaussian mean vectors from its M-component GMM as below

$$\mathbf{v}_{ip} = \left[\boldsymbol{\mu}_{ip1}, \boldsymbol{\mu}_{ip2}, \cdots, \boldsymbol{\mu}_{ipM} \right] , \tag{4.1}$$

where $\boldsymbol{\mu}_{ipm}$, $m = 1, 2, \ldots, M$ is the mean vector of the mth Gaussian component[5]. Similarly, a state cluster supervector $\mathbf{m}_i$ is created from the GMM of state cluster i.

STEP 10 : Collect the trigrapheme state supervectors $\{\mathbf{v}_{i1}, \mathbf{v}_{i2}, \ldots, \mathbf{v}_{iN_i}\}$ as well as the state cluster supervector $\mathbf{m}_i$ of cluster i, and derive an eigenbasis from their correlation matrix using *weighted principal component analysis* (WPCA). The correlation matrix is computed as follows:

$$\frac{1}{n_i} \sum_p n_{ip} (\hat{\mathbf{v}}_{ip} - \hat{\mathbf{m}}_i)(\hat{\mathbf{v}}_{ip} - \hat{\mathbf{m}}_i)' , \tag{4.2}$$

where $\hat{\mathbf{v}}_{ip}$ and $\hat{\mathbf{m}}_i$ are the standardized version of $\mathbf{v}_{ip}$ and $\mathbf{m}_i$ that are created by normalizing them with the diagonal covariance matrix; n_{ip} is the frame count of the

[5]Since the mixture weights are still tied among the trigrapheme states in a state cluster, the M Gaussian components in each state can be consistently ordered across all the member states in the cluster to create their supervectors.

trigrapheme state p in cluster i, and $n_i = \sum_p n_{ip}$ is the total frame count of state cluster i.

STEP 11 : Arrange the eigenvectors $\{\hat{\mathbf{e}}_{ik}, k = 1, 2, \ldots, N_i\}$ in descending order of their eigenvalues λ_{ik}, and pick the top K_i (where $K_i \leq N_i$) eigenvectors to represent the eigenspace of state cluster i. These K_i eigenvectors are now called *eigentrigraphemes* of state cluster i. Note that different state clusters may have different numbers of eigentrigraphemes.

4.2.3.2 Estimation of the Eigentrigrapheme Coefficients

After the derivation of the eigentrigraphemes, the supervector $\mathbf{v}_{ip}$ of any trigrapheme state p in cluster i is assumed to lie in the space spanned by the K_i eigentrigraphemes. Thus, we have

$$\mathbf{v}_{ip} = \mathbf{m}_i + \sum_{k=1}^{K_i} w_{ipk}\mathbf{e}_{ik} \ , \tag{4.3}$$

where $\mathbf{e}_{ik}, k = 1, 2, \ldots, K_i$ is the rescaled version of the standardized eigenvector $\hat{\mathbf{e}}_{ik}$; $\mathbf{w}_{ip} = [w_{ip1}, w_{ip2}, \ldots, w_{ipK_i}]$ is the eigentrigrapheme coefficient vector of trigrapheme state p in the trigrapheme state space of cluster i.

The eigentrigrapheme coefficient vector $\mathbf{w}_{ip}$ is estimated by using the previously introduced *penalized maximum-likelihood eigen-decomposition* (PMLED). The eigentrigrapheme modeling procedure stops if either the estimation of eigentrigrapheme coefficients converges or the recognition accuracy of the trained models is maximum on a development data set. Otherwise, the training data are re-aligned with the current models, and the derivation of eigentrigraphemes and the estimation of their coefficients are repeated.

4.3 Experimental Evaluation

The effectiveness of our eigentrigrapheme acoustic modeling method is evaluated on four under-resourced languages of South Africa with the assumption that no phonetic dictionaries are available. Since graphemes are the basic modeling units in grapheme-based modeling, word recognition accuracy is the main metric for the evaluation.

Nonetheless, triphone-based systems were also built with the use of semi-automatically generated phonetic dictionaries so as to benchmark the results of our eigentrigrapheme results (where no dictionaries are used).

4.3.1 The Lwazi Speech Corpus

The Lwazi project was set up to develop a telephone-based speech-driven information system to take advantage of the more and more popular use of telephones in South Africa nowadays. As part of the project, the Lwazi ASR corpus [67] was collected to provide the necessary speech and language resources in building ASR systems for all eleven official languages of South Africa.

The corpus was collected from approximately 200 speakers per language who are all first language speakers. Each speaker produced approximately 30 utterances, of which 16 of them are phonetically balanced read speech and the remainders are elicited short words such as answers to open questions, answers to yes/no questions, spelt words, dates, and numbers. All the data were recorded over a telephone channel and were transcribed only in words. Background noise, speaker noise, and partial words are marked in the orthographic transcriptions.

The Lwazi project also created a 5,000-word pronunciation dictionary for each language [18]. These dictionaries cover the most common words in the language but not all the words appearing in the corpus. Thus, for the phone-based experiments, the *DictionaryMaker* [91] software was used to generate dictionary entries for the words that are not covered by the Lwazi dictionaries. The given Lwazi dictionaries were used as the seed dictionaries[6] for DictionaryMaker to extract grapheme-to-phoneme conversion rules which were then applied to generate a phonetic transcriptions of the uncovered words for each language. The pronunciations suggested by DictionaryMaker were directly used without any modification.

Among the eleven official South African languages, four are chosen for this investigation. We looked at their ranks according to three different criteria:

- the human language technology (HLT) index [85]: the index indicates the total

[6]For Afrikaans, the dictionary available at http://sourceforge.net/projects/rcrl/files/AfrPronDict/ was used together with the Lwazi dictionary as the seed dictionary.

quantity of HLT activity for each language. The higher the index, the greater the HLT development.

- the phoneme recognition accuracy [9]: a higher phone accuracy means a higher rank for the language.
- the amount of training data available [9]: language with more training data will be given a higher rank.

Table 4.1: Ranks of the four chosen South African languages in three aspects: their human language technology (HLT) indices, phoneme recognition accuracies, and amount of training data in the Lwazi corpus. (A smaller value implies a higher rank.)

Language	HLT Rank [85]	Phoneme Recognition [9]	Amount of Data [9]
Afrikaans	1	5	11
SA English	2	11	10
Sesotho	7	7	7
siSwati	9	3	1

Finally, the four languages are chosen because they have a good mix of phoneme accuracies and HLT activities as shown in Table 4.1:

Afrikaans: Afrikaans is a Low Franconian, West Germanic language, which originated from Dutch [92]. It has about 6 million native speakers and is the third largest language in South Africa. It is also spoken in South Africa's neighbouring countries like Namibia, Botswana and Zimbabwe. It has relatively more resources [78], and more ASR related works [19, 46] have been done on it than other languages of South Africa. It is interesting to see that although Afrikaans has the least amount of training data in the corpus, its phoneme recognition result is quite good among the eleven South African languages.

South African (SA) English: SA English is the de facto South African lingua franca. It is spoken by about 3.6 million people in South Africa. SA English evolved from British English but is highly influenced by Afrikaans and the other languages of the country.

Sesotho: Sesotho is a Southern Bantu language, closely related to other languages in the Sotho-Tswana language group. It has about 3.5 million native speakers and is the seventh largest language in South Africa.

siSwati: siSwati is also a Southern Bantu language, closely related to the Nguni language group. It has about 1 million native speakers and is the ninth largest language in South Africa.

Table 4.2: Information on the data sets of four South African languages used in this investigation. (OOV is *out-of-vocabulary*)

Data Set	#Speakers	#Utt.	Dur.(hr)	Vocab	OOV
Afrikaans Training	160	4,784	3.37	1,513	0.00%
Afrikaans Dev.	20	600	—	870	0.89%
Afrikaans Test	20	599	—	876	0.97%
SA English Training	156	4,665	3.98	1,988	0.00%
SA English Dev.	20	581	—	1,104	1.10%
SA English Test	20	597	—	1,169	1.68%
Sesotho Training	162	4,826	5.70	2,360	0.00%
Sesotho Dev.	20	600	—	1,096	1.86%
Sesotho Test	20	601	—	1,089	2.29%
siSwati Training	156	4,643	8.38	4,645	0.00%
siSwati Dev.	20	599	—	1,889	6.14%
siSwati Test	20	596	—	1,851	4.53%

Since the corpus does not define the training, development and test set for each language, we did the partitions ourselves. The data sets used in our experiments are summarized in Table 4.2. It is interesting to see that languages with more training data (in terms of duration) have a higher percentage of out-of-vocabulary words in their test set.

Table 4.3: Perplexities of phoneme and word language models of the four South African languages.

Language	Data Set	Phoneme Perplexity	Word Bigram Perplexity
Afrikaans	Dev.	7.37 (trigram)	12.4
	Test	7.33 (trigram)	11.18
SA English	Dev.	7.50 (trigram)	13.28
	Test	7.76 (trigram)	11.18
Sesotho	Dev.	10.43 (bigram)	19.60
	Test	10.29 (bigram)	19.69
siSwati	Dev.	7.60 (trigram)	12.27
	Test	7.50 (trigram)	10.94

4.3.2 Common Experimental Settings

The first 13 Perceptual Linear Predictive (PLP) coefficients [42] and their first and second order derivatives were used[7]. These 39-dimensional feature vectors were extracted every 10ms over a window of 25ms. Speaker-based cepstral mean subtraction and variance normalization were performed.

The grapheme set and phoneme set of each language are the same as the ones defined in the Lwazi dictionaries. For all systems described below, the transition probabilities of all triphones/trigraphemes of the same base phone/grapheme were tied together. Each triphone/trigrapheme model was a strictly left-to-right 3-state continuous-density hidden Markov model (CDHMM) with a Gaussian mixture density of at most $M = 16$ components per state. In addition, there were a 1-state short pause model and a 3-state silence model whose middle state was tied with the short pause state. Recognition was performed using the HTK toolkit [98] with a beam search threshold of 350. Only the annotated text data in the training set were used to train the corresponding language models. Both phoneme trigram language models and word bigram language models were estimated for the four languages except Sesotho, for which only phoneme bigrams could be reliably trained. Perplexities of the various language models on the development data and test data are shown in Table 4.3.

All system parameters such as the grammar factor, insertion penalty, regularization parameter β, number of GMM components M, number of tied states or state clusters, and so forth were optimized using the respective development data.

4.3.3 Phoneme and Word Recognition Using Triphone HMMs

We first established the triphone-based ASR benchmarks against which the trigrapheme-based models can be checked. Both conventional tied-state triphone HMM modeling and our cluster-based eigentriphone modeling were tried for the four under-resourced languages of South Africa. The number of base phones, the number of cross-word triphones in the training set, the optimal number of tied states in conventional HMM

[7]MFCC and PLP are two widely used feature extraction schemes in ASR. Both of them are based on Cepstral analysis. They are different in the frequency warping methods and the cepstral representation. In this thesis, we would like to cover both feature extraction schemes. As we have already used MFCC in the experiments presented in Chapter 3, we employ PLP to extract the features in this chapter.

Table 4.4: Some system parameters of triphone modeling in the four South African languages.

Language	#Phonemes	#Triphones	#Tied States in Conventional Models	#State Clusters in Eigentriphone Models
Afrikaans	37	5,203	617	332
SA English	44	7,167	988	362
Sesotho	41	4,061	741	624
siSwati	40	5,140	339	250

training, and the optimal number of state clusters in eigentriphone modeling for each language are summarized in Table 4.4.

Table 4.5: Phoneme recognition accuracy (%) of four South African languages. († The benchmark results in [9] used an older version of the Lwazi corpus and how the corpus were partitioned into training, development, and test sets is unknown.)

Language	Benchmark [9]†	Tied-state Triphone		Cluster-based Eigentriphone	
	Flat LM	Flat LM	N-gram LM	Flat LM	N-gram LM
Afrikaans	63.14	59.07	69.73 (trigram)	62.23	72.32 (trigram)
SA English	54.26	45.48	56.58 (trigram)	46.03	57.84 (trigram)
Sesotho	54.79	62.36	67.06 (bigram)	64.08	68.35 (bigram)
siSwati	64.46	64.76	71.45 (trigram)	68.19	74.13 (trigram)

4.3.3.1 Phoneme Recognition Results

Although word recognition accuracy will be the eventual evaluation metric for grapheme modeling, we would also like to report the phoneme recognition baselines of our triphone models for the sake of completeness. Phoneme recognition was performed on each of the four languages using either none or a flat LM as well as using its respective bigram/trigram LM. The results [8] are given in Table 4.5. The following observations can be made:

- Our phoneme recognition results with flat LMs are quite different from those reported in [9]. There may be a few reasons:

[8]The significant test results in the phoneme recognition with respect to Afrikaans, South African English, Sesotho and siSwati are shown in Table B.4, Table B.5 ,Table B.6 and Table B.7 in appendix B respectively.

- To our knowledge, the Lwazi corpus has been evolving, and the corpus we obtained earlier this year is different from the older version used in [9].
- Since there are no official test sets in the corpus, it is hard to compare recognition performance from different research groups.
- Since the data are not manually labelled by professional transcribers, there is no ground truth which the results from different research groups can be compared with.

Thus, it may not be meaningful to compare our phoneme recognition results with others. We believe it is good enough to see that our results are in the same ballpark as the others.

- SA English has substantially lower phoneme recognition accuracy: it is lower than that of the other three languages by more than 10% absolute. Although SA English has a few more phones in its phonetic inventory than the other languages, and significantly more cross-word triphones to model (see Table 4.4), its phoneme trigram perplexity is actually similar to Afrikaans and siSwati. (Only bigram language model can be reliably estimated for Sesotho, and its value is expected to be higher than the phoneme trigram perplexity of the other three languages.) It means that the phoneme trigrams (as well as triphones) of SA English are more unevenly distributed in the training corpus.

  The lower phoneme recognition accuracy of SA English may be simply due to its larger inventory of phones and triphones, making discrimination among them more difficult. Another plausible reason is that SA English is now the de facto lingua franca of South Africa. It is usually the language of choice for communication among people from different regions and ethnic groups of the country including immigrants from China and India. As a consequence, there are more allophonic variations in SA English, making it harder to recognize.

- The training speech data are not phonetically labelled by human transcribers. Instead, their phonetic transcriptions are generated semi-automatically by grapheme-to-phoneme conversion together with a small bootstrapping dictionary. From the big improvement in recognition performance when phoneme language models were used (vs. when no language models were used), we may conclude that

phoneme language models trained from the generated phonetic transcriptions are good enough to improve phoneme recognition significantly.

- Triphone models estimated by our cluster-based eigentriphone modeling method outperform triphone models estimated by conventional tied-state HMM training by an average of 6.19% relative over the four languages.

Table 4.6: Word recognition accuracy (%) of four South African languages.

Language	Tied-state		Cluster-based Eigen-	
	Trigrapheme	Triphone	Trigrapheme	Triphone
Afrikaans	89.39	89.73	89.87	90.73
SA English	78.30	83.12	79.57	83.72
Sesotho	75.67	75.57	76.35	76.77
siSwati	80.04	79.79	80.67	80.29

4.3.3.2 Word Recognition Results

The word recognition performance [9] of the triphone-based systems are shown in Table 4.6. We can see that

- With no surprise, Sesotho, having the highest LM perplexity (see Table 4.3), has the lowest recognition accuracy.

- For the other three languages, namely Afrikaans, SA English, and siSwati, which all have similar word bigram perplexity, their word recognition performance is well correlated with their vocabulary size and OOV figure. Afrikaans has the best word recognition accuracy, and yet there are only 1,513 words in its vocabulary with 0.97% OOV. On the other hand, siSwati has the worst performance, and its vocabulary size is 4,645 with 4.53% OOV, which are 3–4 times of that of Afrikaans (see Table 4.4).

- Although SA English has the poorest phoneme recognition accuracy, its word recognition performance is second among the four languages. It not only shows

[9] The significant test results in the word recognition with respect to Afrikaans, South African English, Sesotho and siSwati are shown in Table B.8, Table B.9 ,Table B.10 and Table B.11 in appendix B respectively.

the limitations of using phoneme recognition accuracy to predict word recognition performance, but also the effectiveness of a good n-gram language model for word recognition.

- Cluster-based eigentriphone modeling outperforms conventional tied-state HMM training by an average of 5.17% relative over the four languages.

Table 4.7: Some system parameters used in trigrapheme modeling of the four South African languages. (The numbers of possible base graphemes are 43, 26, 27, 26 for the four languages but not all of them are seen in the corpus.)

Language	#Seen Base Graphemes	#Cross-word Trigraphemes	#Tied States in Conventional Models	#State Clusters in Eigentrigrapheme Models
Afrikaans	31	3,458	728	332
SA English	26	4,125	1,630	547
Sesotho	25	3,072	543	543
siSwati	25	3,826	392	255

4.3.4 Word Recognition Using Trigrapheme HMMs

Similar acoustic models were developed using trigraphemes; there is no need for a phonetic dictionary in the process. The number of base graphemes actually observed in the corpus, the number of cross-word trigraphemes in the training set, the optimal number of tied states in conventional HMM training, and the optimal number of state clusters in eigentrigrapheme modeling for each language are summarized in Table 4.7. The word recognition results of the various trigrapheme-based systems are shown in Table 4.6 together with the results from the corresponding triphone-based systems so they can be easily compared.

Besides the observations mentioned in triphone-based systems in Section 4.3.3.2, the following additional observations are well noted.

- Except for SA English, our trigrapheme-based systems performs basically the same as their triphone-based counterparts even without the knowledge of a phonetic dictionary. In fact, trigrapheme-based systems even outperform their triphone-based counterpart in siSwati though insignificantly. The results suggest that there

is a consistent mapping between the pronunciation of Afrikaans, Sesotho, and siSwati and their graphemes.

- Trigrapheme-based systems perform much worse than triphone-based systems in SA English. This is expected as similar results have been reported for English [89]. Besides, as mentioned in Section 4.1, that the pronunciation of English has developed away from its written form over time, the particularly large allophonic variations in SA English (which is also reflected in its phoneme recognition accuracy) further compromise word recognition efforts.

- Once again, our new cluster-based eigentrigrapheme modeling consistently performs better than conventional tied-state trigrapheme HMM training. It has an average gain of 4.08% relative over the four languages.

4.4 Conclusions on Eigentrigrapheme Acoustic Modeling

Most state-of-the-art automatic speech recognition (ASR) systems are developed using phonetic acoustic models. However, for many developing or under-developed countries in the world, the adoption of human language technologies is lagging behind owing to the lack of speech and language resources, which are usually costly and take a lot of human expertise to acquire. Graphemic acoustic modeling mitigates the problem as it does not require a phonetic dictionary. In this chapter, we investigate the use of distinct acoustic modeling on grapheme-based modeling with our proposed method named *cluster-based eigentrigrapheme acoustic modeling*. Our method has the following favorable properties:

- Since our method uses graphemes as the modeling units, it enjoys the same benefits that other grapheme-based modeling methods do. Most importantly, there is no need to create a phone set and a pronunciation dictionary. Thus, it is more favorable for building an ASR system for under-resourced languages.

- Eigentrigrapheme modeling will also enjoy the same benefits as eigentriphone modeling: Many trigraphemes in under-resourced languages may have little training data; in the past, the problem has mainly been solved by state tying,

but eigentrigrapheme modeling allows reliable estimation of the infrequently occurring trigraphemes by careful state clustering and then projecting the member states of each cluster onto a low-dimensional subspace spanned by a small set of eigentrigraphemes of the cluster.

- No language-specific knowledge is required and the whole method is data-driven. It can be used to improve existing systems that are based on conventional tied-state trigrapheme HMMs. In fact, one may implement our method as a post-processing procedure on conventional tied-state trigrapheme HMMs.

- If trigrapheme state clusters are created using the graphemic decision tree, the decision tree may also be used to synthesize unseen trigraphemes in the test lexicon.

For four under-resourced languages of South Africa (SA), namely, Afrikaans, SA English, Sesotho, and siSwati, it is shown that trigrapheme acoustic models trained by our new eigentrigrapheme modeling method consistently outperform the trigrapheme models trained by conventional tied-state HMM training, achieving a relative reduction in the word error rates of the four SA languages by an average of 4.08%. Trigrapheme HMM states trained by the eigentrigrapheme modeling method are distinct from each other — the quantization error among the member states of a tied state in conventional HMM is avoided — and should be more discriminative.

CHAPTER 5

REFERENCE MODEL WEIGHTING

In Chapter 3, we present a new distinct acoustic modeling framework named *eigentriphone* modeling. In *eigentriphone* modeling, an orthogonal eigenbasis is derived using weighted PCA, then all the triphones / triphone states are projected as distinct points onto the space spanned by its eigenvectors. In this chapter, another distinct acoustic modeling method named *reference model weighting* (RMW)[1] is presented. In contrast to eigentriphone modeling, reference model weighting does not require an orthogonal basis, instead, it directly uses a set of reference model vectors in a cluster as the basis. Thus the PCA component can be removed and the preparation of bases is simpler. This chapter starts with the motivation and then the training procedure of reference model weighting. After that, experiments on Wall Street Journal read speech recognition and Switchboard conversational speech recognition are given.

5.1 Motivation from Reference Speaker Weighting

Reference model weighting (RMW) [12] is another attempt at distinct acoustic modeling which is inspired by reference speaker weighting [65] in speaker adaptation. In reference speaker weighting, a set of reference models is employed and the target speaker model is constructed by an interpolation of these reference models. Similarly, in RMW, a set of reference models is employed and the triphone states are constructed by interpolating these reference models. In contrast to eigentriphone modeling where a set of orthogonal basic vectors is derived using PCA, RMW directly use the triphone states as the reference mdoels. Thus, the PCA component for generating orthogonal basic vectors may be omitted, making the training process faster. In fact, the difference between eigentriphone modeling and reference model weighting in distinct acoustic modeling is very similar to the difference between eigenvoice and reference speaker weighting in speaker adaptation.

[1]The first work of RMW is done by my colleague, Dongpeng Chen, and he has already published his work of RMW in a conference [12] before the completion of this thesis.

5.2 The Training Procedure of Reference Model Weighting

RMW follows the overall framework of eigentriphone modeling (ETM) except that it simplifies the training procedure by directly using the triphone states as the reference models. The training procedure of cluster-based RMW is described below. Firstly, decision-tree state clustering is used to construct a set of conventional tied-state triphone HMMs where each state is represented by an M-component GMM. Then the following procedure is repeated for each state cluster i, consisting of N_i members:

STEP 1 : Untie the Gaussian means of all the triphone states in a state cluster with the exception of the unseen triphone states. The means of the cluster GMM are then cloned to initialize *all* untied triphone states in the cluster. Note that the Gaussian covariances and mixture weights of the states in the cluster are still tied together.

STEP 2 : Re-estimate only the Gaussian means of triphone states after cloning. Their Gaussian covariances and mixture weights remain unchanged as their state cluster GMM does.

STEP 3 : Create a triphone state supervector $\mathbf{v}_{ip}$ for each triphone state p in state cluster i by stacking up all its Gaussian mean vectors from its M-component GMM as below

$$\mathbf{v}_{ip} = \left[\boldsymbol{\mu}_{ip1},\, \boldsymbol{\mu}_{ip2},\, \cdots,\, \boldsymbol{\mu}_{ipM} \right] , \tag{5.1}$$

where $\boldsymbol{\mu}_{ipm},\ m = 1, 2, \ldots, M$ is the mean vector of the mth Gaussian component. Similarly, a state cluster supervector $\mathbf{m}_i$ is created from the GMM of state cluster i.

STEP 4 : Arrange the triphone state supervectors $\{\mathbf{v}_{i1},\, \mathbf{v}_{i2},\, \ldots,\, \mathbf{v}_{iN_i}\}$ in descending order of their soft occupation count $\sum_{m,t} \gamma_{ipm}(t)$, and pick the top K_i (where $K_i \leq N_i$) supervectors to represent the space of state cluster i. We called these supervectors the reference state supervectors. Note that different state clusters may have different K_i.

STEP 5 : The supervector $\mathbf{v}_{ip}$ of any triphone state p is assumed to lie in the space

spanned by the K_i reference state supervectors. Thus, we have

$$\mathbf{v}_{ip} = \mathbf{m}_i + \sum_{k=1}^{K_i} w_{ipk}\mathbf{v}_{ik} \ , \tag{5.2}$$

where $\mathbf{w}_{ip} = [w_{ip1}, w_{ip2}, \ldots, w_{ipK_i}]$ is the weight vector of triphone state p..

STEP 6 : Estimate the weight vector $\mathbf{w}_{ip}$ of any triphone state p by PMLED.

The following regularizer is used in RMW for the estimation of weight vector:

$$R(\mathbf{w}_{ip}) = \sum_{k=1}^{N_i} \frac{w_{ipk}^2}{\sum_{m,t} \gamma_{ikm}(t)} \ . \tag{5.3}$$

where $\sum_{m,t} \gamma_{ikm}(t)$ is the sum of occupation counts of all the mixture components of reference model k of cluster i. As a result, the less reliable reference model (with smaller occupation count) are automatically de-emphasized.

5.3 Experiment Evaluation on WSJ: Comparison of RMW and ETM

5.3.1 Experimental Setup

The WSJ recognition task with the SI-84 training set was used for performance comparison of RMW and ETM. Evaluation was performed on the standard Nov92 5K non-verbalized test set, and the si_dt_05 data set was used as the development set for tuning system parameters such as the regularization parameter and decoding parameters, as well as for finding the optimal state-tying nodes and state clusters. Finally, a bigram language model (LM) with a perplexity of 147 was employed in this recognition task.

An HMM system with 1254 tied-states and 32 Gaussian mixtures per state was trained from the training set. All the acoustic models in this task were trained using the HTK toolkit.

5.3.2 Result and Discussion

The comparison of RMW and ETM using different proportions of reference states or eigentriphones is shown in Fig.5.1. The results of the best RMW and ETM systems

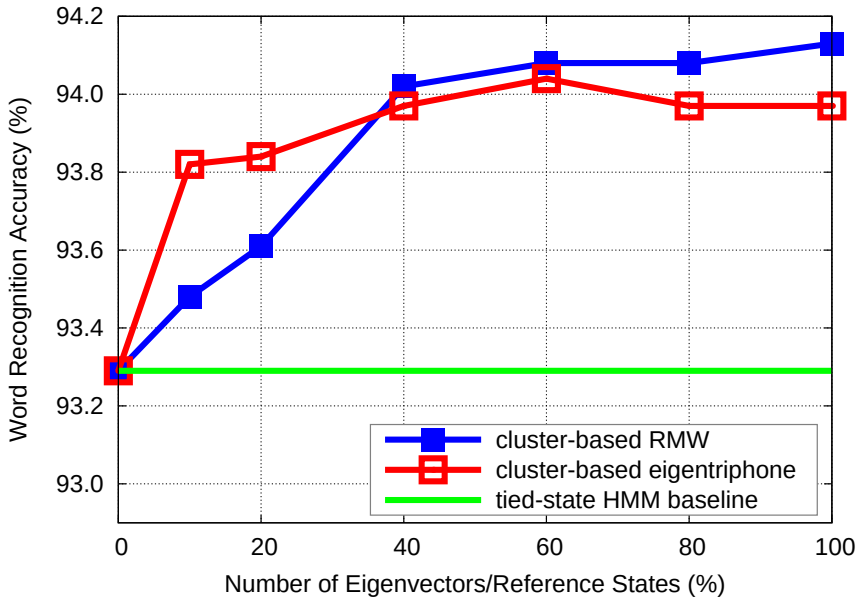

Figure 5.1: Comparison between RMW and ETM when different proportions of reference states or eigentriphones are used on WSJ0.

Table 5.1: Word recognition accuracies (%) and relative Word Error Rate (WER) reduction (%) w.r.t. the tied-state HMM baseline system of various systems on WSJ Nov'92 task.

Model	Word Acc.	WER Reduction	Eigenvectors/Reference States
tied-state HMM	93.29%	-	-
ETM	94.04%	11.18%	60%
RMW	94.13%	12.52%	100%

are summarized in Table 5.1. From the results, we can see that

- Both RMW and ETM show improvement over conventional tied-state triphone system and the best result from RMW is slightly better.
- The performance of RMW decreases when fewer reference states were used and its best performance is achieved when all the reference states were used. In contrast, ETM usually needs pruning some eigenvectors to obtain its best performance.
- ETM has better performance when the percentage of reference states/eigentriphones is lower than 50%. Thus, although RMW is easier to implement, ETM may be a better choice when the model size is a concern.

5.4 Experimental Evaluation on SWB: Performance of RMW together with Other Advanced ASR Techniques

As discussed in Chapter 2, different ASR techniques are proposed in the past to improve the ASR performance in different aspects. For example, feature-based methods are used to improve the robustness of the feature vectors; speaker adaptation methods address the acoustic variations among different speakers; discriminative training improves the performance by formulating an objective function that is more related to the evaluation criteria. RMW, as a kind of distinct acoustic modeling, aims to increase the resolution of the acoustic model by giving each distinct modeling unit a unique model. Thus, it is important to evaluate RMW with other ASR techniques.

In this section, we would like to evaluate the performance of RMW with the presence of other advanced techniques which have been widely used in ASR. Experimental evaluation is done on a conversational speech corpus named *Switchboard* [38]. Besides the speaker adaptation techniques SAT and FMLLR which are discussed in Section 2.3.2, two feature-level techniques are used:

- Linear discriminant analysis (LDA) [10, 28] is commonly used as a feature-space transformation to improve the separability of acoustic classes in the feature space. Consecutive static features in a context window of several frames

are concatenated to form a longer feature vector. Then LDA is applied to reduce the dimensionality of the features. In order to train the LDA transform, a primary acoustic model is needed to classify the feature vectors into classes. [21] reported that using LDA features followed by a maximum likelihood linear transform (MLLT) [39] to replace the conventional dynamic features [29][2] can yield significant performance gain.

- Maximum likelihood linear transform (MLLT) [34, 35] is used to decorrelate different dimensions of the feature vectors. As discussed in Section 2.1.3, Gaussian mixture models with diagonal covariance matrices are often used as the pdfs because of their low computational cost. [35] pointed out that it is preferable to decorrelate the feature vectors so that multiple components can model the possible non-Gaussian distributions rather than the correlation between different dimensions. It is a matrix to rotate the axes so that the correlation between different dimensions of the feature vectors is minimized. It was first proposed to be a model-space transformation so that full covariance matrices are simulated through a transform from diagonal covariance matrices. It can also be implemented as a feature-space transformation where an equivalent effect is obtained. In practice, MLLT is usually used together with LDA features as the MLLT can be integrated with the LDA transform into a single transform.

5.4.1 Speech Corpus and Experimental Setup

A 100-hour training set from Switchboard I [38] was used for the acoustic model estimation [3]. The 100-hour training set contains 76,615 utterances. Recognition results are reported on the standard Hub5 2000 evaluation set. It consists of 1,831 Switchboard utterances and 2,628 Callhome utterances in about 2 hours of conversational speech.

MFCC with cepstral mean and variance normalization were used. Either the dynamic features or LDA features were used to form the acoustic feature vectors. When the dynamic features were used, static features were concatenated with their dynamic coefficients, delta and delta delta, to form 39-dimensional feature vectors. For the LDA

[2]Dynamic features are obtained by taking first and second difference of static features in a neighborhood around the current frame.

[3]The partition of the training set, the number of tied-states and Gaussians and the decoding parameters were suggested by the Switchboard recipe in the Kaldi toolkit.

features, seven consecutive feature vectors, each consisting of 13 static MFCC coefficients, were concatenated and then reduced to 40-dimensional feature vectors using LDA [40]. Then an MLLT [35] was estimated and combined with the LDA transform. SAT [2] and FMLLR [30] were applied on several baselines.

A trigram language model was trained on all the transcribed data of Switchboard I (about 284 hours) using the SRILM toolkit [87]. Acoustic model estimation and recognition were performed using the Kaldi toolkit [25]. Scoring was done by the NIST Scoring toolkit.

The following 4 baseline systems, each with 3K tied-states and 100K Gaussians in total, were trained:

- Baseline1: ML-trained tied-state triphones, 39-dimensional dynamic features
- Baseline2: ML-trained tied-state triphones, 40-dimensional LDA+MLLT features
- Baseline3: ML-trained tied-state triphones, 40-dimensional LDA+MLLT features, SAT and FMLLR applied
- Baseline4: MPE-trained tied-state triphones, 40-dimensional LDA+MLLT features, SAT and FMLLR applied

Cluster-based RMW were applied on each of the above 4 baselines. The training procedures of RMW in this set of experiments is the same as the description in the previous section. All the states were used as reference states to form the basis. The regularization parameter is 400,000 [4].

5.4.2 Result and Discussion

Word recognition results of various systems are summarized in Table 5.2. First of all, we can see that cluster-based RMW outperforms conventional tied-state triphones in all the baseline systems. Furthermore, we find that

[4]This is a typical value for RMW from our experience in the WSJ task.

Table 5.2: Recognition word accuracy (%) of various systems on the Hub5 2000 evaluation set using a trigram language model. The systems were trained on the 100-hour SWB training set. All the systems have around 3K tied-states and 100K Gaussians in total. The numbers in the brackets are the accuracy differences between the RMW systems and their corresponding tied-state systems.

Model Description	Tied-state	RMW
Baseline1: ML training, dynamic features	58	59.4 (+1.4)
Baseline2: ML training, LDA+MLLT features	61.7	62.8 (+1.1)
Baseline3: ML training, LDA+MLLT features, SAT + FMLLR	66.7	67.5 (+0.8)
Baseline4: MPE training, LDA+MLLT features, SAT + FMLLR	69.5	70.1 (+0.6)

- Not surprising, a significant gain is obtained when each ASR technique is added. First of all, there is an absolute gain of 3.7% when dynamic features are replaced by LDA-MLLT features. Then the speaker adaptation techniques SAT and FMLLR obtain another absolute gain of 5%. Finally, MPE discriminative training further obtains an absolute gain of 2.8%.

- Cluster-based RMW can improve all the baseline systems where different kinds of ASR techniques are applied. This suggests that the gain obtained by distinct acoustic modeling is supplementary to the performance of existing state-of-the-art ASR techniques.

- All the RMW systems are statistically and significantly better than the corresponding baseline systems. The significant tests were done by the tool included in the NIST scoring toolkit. The test results are shown in Table B.12 in Appendix B.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This thesis addresses the quantization errors induced by the conventional parameter tying methods. We try to solve the problem by distinct acoustic modeling where every modeling unit has a unique model and a distinct acoustic score. Motivated by the eigenvoice [55] speaker adaptation, we propose a new modeling method called *eigentriphone* modeling. Our new method can robustly estimate rarely occurring triphones without requiring state tying so that all trained triphones are generally distinct from each other. Three variants of the method are investigated, namely the model-based, state-based, and cluster-based eigentriphone modeling. The three variants differ in the modeling unit (triphones or triphone states) and resolution. Empirically we find that the more general cluster-based eigentriphone modeling gives the best performance in both TIMIT phoneme recognition and WSJ word recognition. Cluster-based eigentriphone modeling does not require any modification in the tied-state GMM-HMM training procedures. Thus, our method can be viewed as a kind of post-processing that can easily fit into most existing ASR systems.

Another advantage of our proposed method is that the framework is very flexible and can be applied to any group of modeling unit provided that they may be represented by vectors of the same dimension. In order to test the flexibility of our method, we apply our distinct acoustic modeling framework to grapheme-based modeling systems and the new method is called *eigentrigrapheme* modeling. We show that cluster-based eigentrigrapheme modeling also outperforms conventional tied-state trigrapheme modeling.

We also evaluate another distinct acoustic modeling method named reference model weighting (RMW) which is motivated by reference speaker weighting in speaker adaptation. In contrast to eigentriphone modeling, reference model weighting does not require an orthogonal basis, instead, it directly uses a set of reference model vectors in a cluster as the basis. We show that reference model weighting, using a simpler training procedure, works as well as eigentriphone modeling. We also show that its per-

formance gain is supplementary to the performance of existing state-of-the-art ASR techniques.

6.1 Contributions of the Thesis

The contributions of this thesis are summarized as follows:

- We show that the estimation of triphone models can be formulated as an adaptation problem with our proposed modeling framework called *eigentriphone* modeling. This initiates a new research direction for estimating model parameters.
- We show that state tying is not necessary in modern speech recognition systems. In addition, better performance could be obtained by the use of distinct acoustic modeling.
- We show that our proposed eigentriphone modeling framework is flexible enough that it can be applied to other modeling units.
- We show that the performance gain from the use of distinct acoustic modeling is supplementary to the performance gains from existing ASR techniques.

6.2 Future Work

In the future, we would like to extend our work in the following aspects:

- ***Variance adaptation:*** In our current work, only Gaussian means are adapted whereas the variances are still tied with other members in the same triphone cluster. The major reason is that from the literature [43] Gaussian means are more important to the performance of the acoustic model. Besides, adapting variances in our proposed distinct modeling framework is difficult because of the following reasons:
  - Data sparseness: Since estimating second-order statistics requires much more data than the estimation of first-order statistics, it is difficult to robustly estimate the variance of a triphone with very low occurrence count using only its own data.

- Optimization problem: Unlike the Gaussian mean adaptation, there is no close-form solution in solving the estimation objective function when variance adaptation are taken into consideration. Although optimization methods like gradient descent [5] can be used, they need further hand-tuning to ensure the robustness of the estimation.

A possible solution is to adopt a partial distinct modeling approach where only the triphones with occurrence counts higher than a threshold are adapted and the rest remain tied. One may further employ the maximum likelihood linear regression (MLLR) [32] approach to transform the variances.

- ***Efficient cluster definition:*** In this thesis, we investigate the use of cluster-based eigentriphone modeling on state clusters defined by the conventional tree-based state clustering. This tree-based state clustering defines the state clusters by maximizing the likelihood of the data whereas the eigentriphone modeling aims at representing the reference models with as few eigenvectors as possible. Thus, there is a mismatch between the objectives of two approaches. It is possible to eliminate the mismatch by using another objective function in the tree-based state clustering. The question is how to measure the effectiveness of the new state definition. One possible way is to maximize the total area under the eigen spectrums of every state clusters.

- ***Phonetic units with higher levels of context dependence:*** One of the major advantages of the eigentriphone modeling framework is that it is very flexible and can be applied to other modeling unit. We have successfully applied the framework on a grapheme-based modeling system and we called it eigentrigrapheme modeling. One may further extend the framework to estimate phonetic units with higher levels of context dependence such as quinphones [95] or pentaphones [71].

- ***Distinct triphone states in DNN-HMM:*** One of the major findings in this thesis is another option of context-dependent model estimation apart from parameter tying. In this thesis, we demonstrate the use of distinct acoustic modeling in the GMM-HMM framework and it is possible to extend the idea to other recognizer frameworks. As mentioned in Section 2.1.3, DNN-HMM has received a

lot of attention recently; the output nodes of a DNN usually represent the posterior probabilities of the tied states of an HMM [45]. It will be interesting to see how it performs when the tied states are replaced by distinct triphone states. The risk of overfitting due to the increased number of output nodes and connection weights in DNN need to be addressed. One possible direction is to try the recently proposed "dropout" [15] method in the DNN-HMM framework. The dropout procedure randomly omits each hidden unit according to a probability distribution and the resulting DNN is approximately an average of multiple neural networks. As the training cases for infrequent triphones are limited, averaging the connection weights can help to avoid overfitting. Dropout has already been used successfully in improving low-resource ASR where training data is limited [68].

APPENDIX A

PHONE SET IN THE THESIS

Table A.1: The phone set and their examples.

Phoneme	Example	Transcription
aa	ODD	aa d
ae	AT	ae t
ah	HUT	hh ah t
ao	OUGHT	ao t
aw	COW	k aw
ay	HIDE	hh ay d
b	BE	b iy
ch	CHEESE	ch iy z
d	END	eh n d
dh	WEATHER	w eh dh er
eh	BEAR	b eh r
er	HURT	hh er t
ey	ATE	ey t
f	FREE	f r iy
g	GREEN	g r iy n
hh	HE	hh iy
ih	IT	ih t
iy	EAT	iy t
jh	JANE	jh ey n
k	KEY	k iy
l	LIGHT	l ay t
m	ME	m iy
n	SON	s ah n
ng	PING	p ih ng
ow	NO	n ow
oy	TOY	t oy
p	PIG	p ih g
r	RIGHT	r ay t
s	SEA	s iy
sh	SHE	sh iy
t	TEA	t iy
th	THETA	th ey t ah
uh	FOOT	f uh t
uw	TWO	t uw
v	VERY	v eh r iy
w	WET	w eh t
y	YET	y eh t
z	ZOO	z uw
zh	VISION	v ih zh ah n

APPENDIX B

SIGNIFICANT TESTS

The statistical significance test suite from National Institute of Standards and Technology (NIST) is used to compared different systems. It encompasses four tests:

- **MP:** Matched Pair Sentence Segment (Word Error) Test.
- **SP:** Signed Paired Comparison (Speaker Word Accuracy Rate) Test.
- **WI:** Wilcoxon Signed Rank (Speaker Word Accuracy Rate) Test.
- **MN:** McNemar (Sentence Error) Test.

Here, we first apply the test to compare our proposed cluster-based eigentriphone modeling and the conventional state tying. The abbreviations of the two system are as follows:

- **Cluster-ETM:** Cluster-based eigentrphone modeling.
- **Tied-state:** Conventional state tying.

The test results are shown in Table B.1, Table B.2 and Table B.3.

Table B.1: Significant tests of the TIMIT experiments.

	Cluster-ETM
Tied-state	MP: Cluster-ETM SP: same WI: same MN: same

Table B.2: Significant tests of the WSJ nov92 experiments.

	Cluster-ETM
Tied-state	MP: same SP: same WI: same MN: same

Table B.3: Significant tests of the WSJ nov93 experiments.

	Cluster-ETM
Tied-state	MP: same SP: same WI: same MN: same

Next, we apply the test to compare various systems we used in Chapter 4. The abbreviations of various systems in the phoneme recognition are summarized as follows:

- **TIED-FLATLM:** Phone-based system with conventional state tying and a flat LM used.
- **CETM-FLATLM:** Cluster-based eigentriphone modeling with a flat LM used.
- **TIED-BILM:** Phone-based system with conventional state tying and a bigram LM used.
- **CETM-BILM:** Cluster-based eigentriphone modeling with a bigram LM used.
- **TIED-TRILM:** Phone-based system with conventional state tying and a trigram LM used.
- **CETM-TRILM:** Cluster-based eigentriphone modeling with a trigram LM used.

The test results with respect to Afrikaans, South African English, Sesotho and siSwati are shown in Table B.4, Table B.5 ,Table B.6 and Table B.7 respectively.

The abbreviations of various systems in the word recognition are summarized as follows:

- **PHONE-TIED:** Phone-based system with conventional state tying.

- **GRAPHEME-TIED:** Grapheme-based system with conventional state tying.
- **PHONE-CETM:** Cluster-based eigentriphone modeling.
- **GRAPHEME-CETM:** Cluster-based eigentrigrapheme modeling.

The test results with respect to Afrikaans, South African English, Sesotho and siSwati are shown in Table B.8, Table B.9 ,Table B.10 and Table B.11 respectively.

Table B.4: Significant tests of the Afrikaans phoneme recognition experiments.

	CETM-FLATLM	TIED-TRILM	CETM-TRILM
TIED-FLATLM	MP: CETM-FLATLM SP: same WI: same MN: CETM-FLATLM	MP: TIED-TRILM SP: same WI: same MN: TIED-TRILM	MP: CETM-TRILM SP: same WI: same MN: CETM-TRILM
CETM-FLATLM		MP: TIED-TRILM SP: same WI: same MN: TIED-TRILM	MP: CETM-TRILM SP: same WI: same MN: CETM-TRILM
TIED-TRILM			MP: CETM-TRILM SP: same WI: same MN: same

Table B.5: Significant tests of the SA English phoneme recognition experiments.

	CETM-FLATLM	TIED-TRILM	CETM-TRILM
TIED-FLATLM	MP: CETM-FLATLM SP: same WI: same MN: same	MP: TIED-TRILM SP: same WI: same MN: TIED-TRILM	MP: CETM-TRILM SP: same WI: same MN: CETM-TRILM
CETM-FLATLM		MP: TIED-TRILM SP: same WI: same MN: TIED-TRILM	MP: CETM-TRILM SP: same WI: same MN: CETM-TRILM
TIED-TRILM			MP: CETM-TRILM SP: same WI: same MN: same

Table B.6: Significant tests of the Sesotho phoneme recognition experiments.

	CETM-FLATLM	TIED-BILM	CETM-BILM
TIED-FLATLM	MP: CETM-FLATLM SP: same WI: same MN: same	MP: TIED-BILM SP: same WI: same MN: same	MP: CETM-BILM SP: same WI: same MN: same
CETM-FLATLM		MP: TIED-BILM SP: same WI: same MN: same	MP: CETM-BILM SP: same WI: same MN: same
TIED-BILM			MP: CETM-BILM SP: same WI: same MN: same

Table B.7: Significant tests of the siSwati phoneme recognition experiments.

	CETM-FLATLM	TIED-TRILM	CETM-TRILM
TIED-FLATLM	MP: CETM-FLATLM SP: same WI: same MN: same	MP: TIED-TRILM SP: same WI: same MN: TIED-TRILM	MP: CETM-TRILM SP: same WI: same MN: CETM-TRILM
CETM-FLATLM		MP: TIED-TRILM SP: same WI: same MN: same	MP: CETM-TRILM SP: same WI: same MN: CETM-TRILM
TIED-TRILM			MP: CETM-TRILM SP: same WI: same MN: CETM-TRILM

Table B.8: Significant tests of the Afrikaans word recognition experiments.

	PHONE-TIED	GRAPHEME-CETM	PHONE-CETM
GRAPHEME-TIED	MP: same SP: same WI: same MN: same	MP: same SP: same WI: same MN: same	MP: PHONE-CETM SP: same WI: same MN: same
PHONE-TIED		MP: same SP: same WI: same MN: same	MP: PHONE-CETM SP: same WI: same MN: PHONE-CETM
GRAPHEME-CETM			MP: PHONE-CETM SP: same WI: same MN: PHONE-CETM

Table B.9: Significant tests of the SA English word recognition experiments.

	PHONE-TIED	GRAPHEME-CETM	PHONE-CETM
GRAPHEME-TIED	MP: PHONE-TIED SP: same WI: same MN: PHONE-TIED	MP: same SP: same WI: same MN: same	MP: PHONE-CETM SP: same WI: same MN: PHONE-CETM
PHONE-TIED		MP: PHONE-TIED SP: same WI: same MN: PHONE-TIED	MP: same SP: same WI: same MN: same
GRAPHEME-CETM			MP: PHONE-CETM SP: same WI: same MN: PHONE-CETM

Table B.10: Significant tests of the Sesotho word recognition experiments.

	PHONE-TIED	GRAPHEME-CETM	PHONE-CETM
GRAPHEME-TIED	MP: same SP: same WI: same MN: same	MP: same SP: same WI: same MN: same	MP: PHONE-CETM SP: same WI: same MN: same
PHONE-TIED		MP: same SP: same WI: same MN: same	MP: PHONE-CETM SP: same WI: same MN: same
GRAPHEME-CETM			MP: same SP: same WI: same MN: same

Table B.11: Significant tests of the siSwati word recognition experiments.

	PHONE-TIED	GRAPHEME-CETM	PHONE-CETM
GRAPHEME-TIED	MP: same SP: same WI: same MN: same	MP: same SP: same WI: same MN: GRAPHEME-CETM	MP: same SP: same WI: same MN: same
PHONE-TIED		MP: GRAPHEME-CETM SP: same WI: same MN: same	MP: same SP: same WI: same MN: same
GRAPHEME-CETM			MP: same SP: same WI: same MN: same

Finally, we apply the test to compare various systems we used in the Switchboard experiments in Chapter 5. The abbreviations of the systems are as follows:

- **ML-TS:** Conventional state tying with ML training and dynamic features.
- **ML-RMW:** RMW with ML training and dynamic features.
- **ML-LDA-TS:** Conventional state tying with ML training and LDA+MLLT features.
- **ML-LDA-RMW:** RMW with ML training and LDA+MLLT features.
- **ML-LDA-SAT-TS:** Conventional state tying with ML training, LDA+MLLT features and SAT+FMLLT applied.
- **ML-LDA-SAT-RMW:** RMW with ML training, LDA+MLLT features and SAT+FMLLT applied.
- **MPE-LDA-SAT-TS:** Conventional state tying with MPE training, LDA+MLLT features and SAT+FMLLT applied.
- **MPE-LDA-SAT-RMW:** RMW with MPE training, LDA+MLLT features and SAT+FMLLT applied.

The test results are shown in Table B.12.

Table B.12: Significant tests of the Switchboard experiments.

	ML-RMW	ML-LDA-TS	ML-LDA-RMW	ML-LDA-SAT-TS	ML-LDA-SAT-RMW	MPE-LDA-SAT-TS	MPE-LDA-SAT-RMW
ML-TS	MP: ML-RMW SP: ML-RMW WI: ML-RMW MN: ML-RMW	MP: ML-LDA-TS SP: ML-LDA-TS WI: ML-LDA-TS MN: ML-LDA-TS	MP: ML-LDA-RMW SP: ML-LDA-RMW WI: ML-LDA-RMW MN: ML-LDA-RMW	MP: ML-LDA-SAT-TS SP: ML-LDA-SAT-TS WI: ML-LDA-SAT-TS MN: ML-LDA-SAT-TS	MP: ML-LDA-SAT-RMW SP: ML-LDA-SAT-RMW WI: ML-LDA-SAT-RMW MN: ML-LDA-SAT-RMW	MP: MPE-LDA-SAT-TS SP: MPE-LDA-SAT-TS WI: MPE-LDA-SAT-TS MN: MPE-LDA-SAT-TS	MP: MPE-LDA-SAT-RMW SP: MPE-LDA-SAT-RMW WI: MPE-LDA-SAT-RMW MN: MPE-LDA-SAT-RMW
ML-RMW		MP: ML-LDA-TS SP: ML-LDA-TS WI: ML-LDA-TS MN: ML-LDA-TS	MP: ML-LDA-RMW SP: ML-LDA-RMW WI: ML-LDA-RMW MN: ML-LDA-RMW	MP: ML-LDA-SAT-TS SP: ML-LDA-SAT-TS WI: ML-LDA-SAT-TS MN: ML-LDA-SAT-TS	MP: ML-LDA-SAT-RMW SP: ML-LDA-SAT-RMW WI: ML-LDA-SAT-RMW MN: ML-LDA-SAT-RMW	MP: MPE-LDA-SAT-TS SP: MPE-LDA-SAT-TS WI: MPE-LDA-SAT-TS MN: MPE-LDA-SAT-TS	MP: MPE-LDA-SAT-RMW SP: MPE-LDA-SAT-RMW WI: MPE-LDA-SAT-RMW MN: MPE-LDA-SAT-RMW
ML-LDA-TS			MP: ML-LDA-RMW SP: ML-LDA-RMW WI: ML-LDA-RMW MN: ML-LDA-RMW	MP: ML-LDA-SAT-TS SP: ML-LDA-SAT-TS WI: ML-LDA-SAT-TS MN: ML-LDA-SAT-TS	MP: ML-LDA-SAT-RMW SP: ML-LDA-SAT-RMW WI: ML-LDA-SAT-RMW MN: ML-LDA-SAT-RMW	MP: MPE-LDA-SAT-TS SP: MPE-LDA-SAT-TS WI: MPE-LDA-SAT-TS MN: MPE-LDA-SAT-TS	MP: MPE-LDA-SAT-RMW SP: MPE-LDA-SAT-RMW WI: MPE-LDA-SAT-RMW MN: MPE-LDA-SAT-RMW
ML-LDA-RMW				MP: ML-LDA-SAT-TS SP: ML-LDA-SAT-TS WI: ML-LDA-SAT-TS MN: ML-LDA-SAT-TS	MP: ML-LDA-SAT-RMW SP: ML-LDA-SAT-RMW WI: ML-LDA-SAT-RMW MN: ML-LDA-SAT-RMW	MP: MPE-LDA-SAT-TS SP: MPE-LDA-SAT-TS WI: MPE-LDA-SAT-TS MN: MPE-LDA-SAT-TS	MP: MPE-LDA-SAT-RMW SP: MPE-LDA-SAT-RMW WI: MPE-LDA-SAT-RMW MN: MPE-LDA-SAT-RMW
ML-LDA-SAT-TS					MP: ML-LDA-SAT-RMW SP: ML-LDA-SAT-RMW WI: ML-LDA-SAT-RMW MN: ML-LDA-SAT-RMW	MP: MPE-LDA-SAT-TS SP: MPE-LDA-SAT-TS WI: MPE-LDA-SAT-TS MN: MPE-LDA-SAT-TS	MP: MPE-LDA-SAT-RMW SP: MPE-LDA-SAT-RMW WI: MPE-LDA-SAT-RMW MN: MPE-LDA-SAT-RMW
ML-LDA-SAT-RMW						MP: MPE-LDA-SAT-TS SP: MPE-LDA-SAT-TS WI: MPE-LDA-SAT-TS MN: MPE-LDA-SAT-TS	MP: MPE-LDA-SAT-RMW SP: MPE-LDA-SAT-RMW WI: MPE-LDA-SAT-RMW MN: MPE-LDA-SAT-RMW
MPE-LDA-SAT-TS							MP: MPE-LDA-SAT-RMW SP: MPE-LDA-SAT-RMW WI: MPE-LDA-SAT-RMW MN: MPE-LDA-SAT-RMW

REFERENCES

[1] T. Anastasakos, J. McDonough, and J. Makhoul. Speaker adaptive training: a maximum likelihood approach to speaker normalization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1043–1046, 1997.

[2] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1137–1140, 1996.

[3] O. Andersen, R. Kuhn, A. Lazarides, P. Dalsgaard, J. Haas, and E. Noth. Comparison of two tree-structured approaches for grapheme-to-phoneme conversion. In *Proceedings of the International Conference on Spoken Language Processing*, 1996.

[4] L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek., B. L. Lewis, and R. L. Mercer. Further results on the recognition of a continuously read natural corpus. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1980.

[5] J. Barzilai and J. Borwein. Two point step size gradient methods. *IMA Journal of Numerical Analysis*, pages 141–148, 1988.

[6] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[7] J.R. Bellegarda. Unsupervised, language-independent grapheme-to-phoneme conversion by latent analogy. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003.

[8] E. Bocchieri and B. Mak. Subspace distribution clustering hidden Markov model. *IEEE Transactions on Speech and Audio Processing*, 9:264–275, 2001.

[9] E. Barnard C. van Heerden and M. Davel. Basic speech recognition for spoken dialogues. In *Proceedings of Interspeech*, 2009.

[10] N. Campbell. *Canonical variate analysis - a general formulation*. Australian Journal of Statistics, 1984.

[11] Hung-An Chang and James R. Glass. A back-off discriminative acoustic model for automatic speech recognition. In *Proceedings of Interspeech*, 2009.

[12] Dongpeng Chen and Brian Mak. Distinct triphone modeling by reference model weighting. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 7150–7153, 2013.

[13] W. Chou, C. H. Lee, and B. H. Juang. Minimum error rate training based on n-best string models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 652–655, 1993.

[14] Y. L. Chow, R. M. Schwartz, S. Roucos, 0. Kimball, P. Price, F. Kubala, M. Dunham, M. Krasner, and J. Makhoul. The role of word-dependent coarticulatory effects in a phoneme-based speech recognition system. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1986.

[15] George E. Dahl, Tara N. Sainath, and Geoffrey E. Hinton. Improving deep neural networks for LVCSR using rectified linear units and dropout. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 8609–8613, 2013.

[16] P.T. Daniels and W. (Eds.) Bright. *The World's Writing Systems*. Oxford University Press, 198 Madison Avenue, New York, NY 10016, USA, 1996.

[17] M. Davel and E. Barnard. The efficient creation of pronuniciation dictionaries: human factors in bootstrapping. In *Proceedings of Interspeech*, 2004.

[18] M. Davel and O. Martirosian. Pronunciation dictionary development in resource-scarce environments. In *Proceedings of Interspeech*, 2009.

[19] Febe de Wet, Alta de Waal, and Gerhard B van Huyssteen. Developing a broad-band automatic speech recognition system for Afrikaans. In *Proceedings of Interspeech*, 2011.

[20] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer. Speaker adaptation using constrained estimation of Gaussian mixtures. *IEEE Transactions on Speech and Audio Processing*, 3(5):357–366, 1995.

[21] H. Erdogan. Regularizing linear discriminant analysis for speech recognition. In *Proceedings of Interspeech*, pages 3021–3024, 2005.

[22] D. Povey et al. Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 4334–4337, 2010.

[23] D. Povey et al. Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010.

[24] D. Povey et al. Subspace Gaussian mixture models for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 4330–4333, 2010.

[25] D. Povey et al. The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

[26] P. Beyerlein et al. Towards language independent acoustic modeling. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 1999.

[27] R. Kuhn et al. Eigenvoices for speaker adaptation. *Proceedings of the International Conference on Spoken Language Processing*, pages 1771–1774, 1998.

[28] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press Professional, 1972.

[29] S. Furui. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(1):52–59, 1986.

[30] M. J. F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 25(2):75–78, 1997.

[31] M. J. F. Gales. Transformation streams and the HMM error model. *Computer Speech and Language*, 2002.

[32] M. J. F. Gales and P. C. Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, 10:249–264, 1996.

[33] M. J. F. Gales and K. Yu. Canonical state models for automatic speech recognition. In *Proceedings of Interspeech*, pages 58–61, 2010.

[34] M.J.F. Gales. Semi-tied covariance matrices. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 657–660, 1998.

[35] M.J.F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 7(3):272–281, 1999.

[36] Jean-Luc Gauvain and Chin-Hui Lee. Bayesian learning of Gaussian mixture densities for hidden Markov models. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 272–277, 1991.

[37] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, pages 291–298, 1994.

[38] J. J. Godfrey, E. C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 517–520, 1992.

[39] R. Gopinath. Maximum likelihood modeling with Gaussian distributions for classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 661–664, 1998.

[40] R. Haeb-Umbach and H. Ney. Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 13–16, 1992.

[41] Tim J. Hazen. A comparison of novel techniques for rapid speaker adaptation. *Speech Communications*, 31:15–33, 2000.

[42] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

[43] X. D. Huang, A. Acero, and H. W. Hon. *Spoken language processing: A guide to theory, algorithm and system development*. Prentice Hall, NY, 2001.

[44] X. D. Huang and M. A. Jack. Semi-continuous hidden Markov models for speech signals. *Computer Speech and Language*, 3:239–251, 1989.

[45] M. Y. Hwang and X. D. Huang. Shared-distribution hidden Markov model for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 1:414–420, 1993.

[46] Herman Kamper and Thomas Niesler. Multi-accent speech recognition of Afrikaans, black and white varieties of South African English. In *Proceedings of Interspeech*, 2011.

[47] S. Kanthak and H. Ney. Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002.

[48] S. Kanthak and H. Ney. Multilingual acoustic modeling using graphemes. In *Proceedings of the European Conference on Speech Communication and Technology*, 2003.

[49] T. Ko and B. Mak. A fully automated derivation of state-based eigentriphones for triphone modeling with no tied states using regularization. In *Proceedings of Interspeech*, pages 781–784, 2011.

[50] T. Ko and B. Mak. Derivation of eigentriphones by weighted principal component analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 4097–4100, 2012.

[51] T. Ko and B. Mak. Eigentrigraphemes for under-resourced languages. *Speech Communications*, 2013.

[52] T. Ko and B. Mak. Eigentriphones for context-dependent acoustic modeling. *IEEE Transactions on Audio, Speech and Language Processing*, 21(6):1285–1294, 2013.

[53] J. Kohler. Multi-lingual phoneme recognition exploiting acoustic-phonetic similarities of sounds. In *Proceedings of the International Conference on Spoken Language Processing*, 1996.

[54] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski. Rapid speaker adaptation in eigenvoice space. *IEEE Transactions on Speech and Audio Processing*, 8(4):695–707, Nov 2000.

[55] R. Kuhn, P. Nguyen, J. C. Junqua, and L. Goldwasser. Eigenfaces and eigenvoices: Dimensionality reduction for specialized pattern recognition. In *Multimedia Signal Processing, 1998 IEEE Second Workshop*, pages 71–76, 1998.

[56] R. Kuhn, F. Perronnin, P. Nguyen, J.C. Junqua, and L. Rigazio. Very fast adaptation with a compact context-dependent eigenvoice model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 373–376, 2001.

[57] Roland Kuhn, Jean-Claude Junqua, Patrick Nguyen, and Nancy Niedzielski. Rapid speaker adaptation in eigenvoice space. *IEEE Transactions on Speech and Audio Processing*, 8:695–707, 2000.

[58] Viet-Bac Le and L. Besacier. Automatic speech recognition for under-resourced languages: Application to vietnamese language. *IEEE Transactions on Audio, Speech and Language Processing*, 17:1471–1482, 2009.

[59] Chin-Hui Lee and Jean-Luc Gauvain. Speaker adaptation based on map estimation of HMM parameters. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 558–561, 1993.

[60] Chin-Hui Lee, Chih-Heng Lin, and Biing-Hwang Juang. A study on speaker adaptation of the parameters of continuous density hidden Markov models. *IEEE Transactions on Signal Processing*, pages 806–814, 1991.

[61] K. F. Lee. *The Development of the SPHINX System*. Kluwer Academic Publishers, 1989.

[62] K. F. Lee. Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 38:599–609, 1990.

[63] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language*, 9:171–186, 1995.

[64] Liang Lu, Arnab Ghoshal, and Steve Renals. Regularized subspace Gaussian mixture models for cross-lingual speech recognition. *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, pages 365–370, 2011.

[65] Brian Mak, Tsz-Chung Lai, and Roger Hsiao. Improving reference speaker weighting adaptation by the use of maximum-likelihood reference speakers. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 229–232, 2006.

[66] H. Meng, S. Hunnicutt, S. Seneff, and V. Zue. Reversible lettter-to-sound generation based on parsing word morphology. *Speech Communications*, 18:47–63, 1996.

[67] Meraka-Institute. *Lwazi ASR corpus*. Online:http://www.meraka.org.za/lwazi, 2009.

[68] Yajie Miao and Florian Metze. Improving low-resource CD-DNN-HMM using dropout and multilingual DNN training. In *Proceedings of Interspeech*, pages 2237–2241, 2013.

[69] N. Morgan and H. Bourlard. Continuous speech recognition using multilayer perceptrons with hidden Markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1990.

[70] N. Morgan and H. Bourlard. Connectionist speech recognition: a hybrid approach. *Springer*, 1994.

[71] J. J. Odell. *The Use of Context in Large Vocabulary Speech Recognition*. Ph.D. Thesis, Cambridge University Engineering Department, England, 1995.

[72] K.U. Ogbureke and J. Carson-Berndsen. Framework for cross-language automatic phonetic segmentation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010.

[73] Sanjika Hewavitharana Paisarn Charoenpornsawat and Tanja Schultz. Thai grapheme-based speech recognition. In *In Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, 2006.

[74] Jia Pan, Cong Liu, Zhiguo Wang, Yu Hu, and Hui Jiang. Investigation of deep neural networks (DNN) for large vocabulary continuous speech recognition: Why DNN surpasses GMMs in acoustic modeling. In *Proceedings of the International Symposium of Chinese Spoken Language Processing*, pages 301–305, 2012.

[75] D. B. Paul and J. M. Baker. The design of the Wall Street Journal-based CSR corpus. In *Proceedings of the DARPA Speech and Natural Language Workshop*, February 1992.

[76] D. Povey and P. C. Woodland. Improved discriminative training techniques for large vocabulary continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 45–48, 2001.

[77] Daniel Povey. *Discriminative Training for Large Vocabulary Speech Recognition*. Ph.D. Thesis, 2003.

[78] J. C. Roux, P. H. Louw, and T. R. Niesler. The African speech technology project: An assessment. In *Proceedings of LREC*, 2004.

[79] J. J. Odell S. J. Young and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the Workshop on Human Language Technology*, 1994.

[80] E. G. Schukat-Talamazzini, H. Niemann, W. Eckert, T. Kuhn, and S. Rieck. Automatic speech recognition without phonemes. In *Proceedings of the European Conference on Speech Communication and Technology*, 1993.

[81] R. M. Schwartz, Y. L. Chow, 0. Kimball, S. Roucos, M. Krasner, and J. Makhoul. Context-dependent modeling for acoustic-phonetic recognition of continuous speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1985.

[82] R. M. Schwartz, Y. L. Chow, S. Roucos, M. Krasner, and J. Makhoul. Improved hidden Markov modeling phonemes for continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1984.

[83] F. Seide, G. Li, X. Chen, and D. Yu. Feature engineering in context-dependent deep neural networks for conversational speech transcription. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, pages 24–29, 2011.

[84] F. Seide, G. Li, and D. Yu. Conversational speech transcription using context-dependent deep neural networks. In *Proceedings of Interspeech*, 2011.

[85] A. Sharma-Grover, G. B. van Huyssteen, and M. W. Pretorius. An HLT profile of the official South African languages. In *Proceedings of the Second Workshop on African Language Technology (AfLaT 2010)*, 2010.

[86] J. J. Sooful and E. C. Botha. An acoustic distance measure for automatic cross-language phoneme mapping. In *Proceedings of Pattern Recognition Association of South Africa*, 2001.

[87] A. Stolcke. SRILM - An extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904, 2002.

[88] Sebastian Stuker. Modified polyphone decision tree specialization for porting multilingual grapheme based asr systems to new languages. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008.

[89] Sebastian Stuker. *Acoustic Modelling for Under-Resourced Languages*. Ph.D. Thesis, 2009.

[90] S. Takahashi and S. Sagayama. Four-level tied-structure for efficient representation of acoustic modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995.

[91] M. Tempest and M.Davel. *DictionaryMaker 2.16 user manual*. http://dictionarymaker.sourceforge.net., 2009.

[92] G. B. van Huyssteen and S. Pilon. Rule-based conversion of closely-related languages: A Dutch-to-Afrikaans convertor. In *Proceedings of the 20th Annual Symposium of the Pattern Recognition Association of South Africa*, 2009.

[93] L. R. Welch. Hidden Markov models and the Baum-Welch algorithm. *IEEE Information Theory Society Newsletter*, 54(4), 2003.

[94] P.C. Woodland. Speaker adaptation for continuous density HMMs: A review. In *ISCAITR Workshop on Adaptation Methods for Speech Recognition*, 2001.

[95] P.C. Woodland and D. Povey. Large scale MMIE training for conversational telephone speech recognition. In *Proceedings of Speech Transcription Workshop*, 2000.

[96] S. J. Young. The general use of tying in phoneme-based HMM speech recognisers. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1992.

[97] S. J. Young and P. C. Woodland. The use of state tying in continuous speech recognition. In *Proceedings of the European Conference on Speech Communication and Technology*, volume 3, pages 2203–2206, 1993.

[98] Steve Young et al. *The HTK Book (Version 3.4)*. University of Cambridge, 2006.

[99] V Zue, S. Seneff, and J. Glass. Speech database development at MIT: TIMIT and beyond. *Speech Communication*, 9(4):351–356, August 1990.