AUTOMATED PHONEME MAPPING FOR CROSS-LANGUAGE SPEECH RECOGNITION

by

Jayren Jugpal Sooful

Submitted in partial fulfillment of the requirements for the degree

Master of Engineering (Computer Engineering)

in the

Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

Pretoria

June 2004

AUTOMATED PHONEME MAPPING FOR CROSS-LANGUAGE SPEECH RECOGNITION

By: Jayren Jugpal Sooful

Supervisor: Prof. Elizabeth C. Botha

Department: Electrical, Electronic and Computer Engineering

Degree: MEng (Computer Engineering)

Summary

This dissertation explores a unique automated approach to map one phoneme set to another, based on the acoustic distances between the individual phonemes. Although the focus of this investigation is on cross-language applications, this automated approach can be extended to same-language but different-database applications as well.

The main goal of this investigation is to be able to use the data of a source language, to train the initial acoustic models of a target language for which very little speech data may be available. To do this, an automatic technique for mapping the phonemes of the two data sets must be found. Using this technique, it would be possible to accelerate the development of a speech recognition system for a new language. The current research in the cross-language speech recognition field has focused on manual methods to map phonemes. This investigation has considered an English-to-Afrikaans phoneme mapping, as well as an Afrikaans-to-English phoneme mapping. This has been previously applied to these language instances, but utilising manual phoneme mapping methods.

To determine the best phoneme mapping, different acoustic distance measures are compared. The distance measures that are considered are the Kullback-Leibler measure, the Bhattacharyya distance metric, the Mahalanobis measure, the Euclidean measure, the L2 metric and the Jeffreys-Matusita distance. The distance measures are tested by comparing the cross-database recognition results obtained on phoneme models created from the TIMIT speech corpus and a locally-compiled South African SUN Speech database. By selecting the most appropriate distance measure, an automated procedure to map phonemes from the source

language to the target language can be done. The best distance measure for the mapping gives recognition rates comparable to a manual mapping process undertaken by a phonetic expert.

This study also investigates the effect of the number of Gaussian mixture components on the mapping and on the speech recognition system's performance. The results indicate that the recogniser's performance increases up to a limit as the number of mixtures increase. In addition, this study has explored the effect of excluding the Mel Frequency delta and acceleration cepstral coefficients. It is found that the inclusion of these temporal features help improve the mapping and the recognition system's phoneme recognition rate. Experiments are also carried out to determine the impact of the number of HMM recogniser states. It is found that single-state HMMs deliver the optimum cross-language phoneme recognition results.

After having done the mapping, speaker adaptation strategies are applied on the recognisers to improve their target-language performance. The models of a fully trained speech recogniser in a source language are adapted to target-language models using Maximum Likelihood Linear Regression (MLLR) followed by Maximum A Posteriori (MAP) techniques. Embedded Baum-Welch re-estimation is used to further adapt the models to the target language. These techniques result in a considerable improvement in the phoneme recognition rate. Although a combination of MLLR and MAP techniques have been used previously in speech adaptation studies, the combination of MLLR, MAP and EBWR in cross-language speech recognition is a unique contribution of this study.

Finally, a data pooling technique is applied to build a new recogniser using the automatically mapped phonemes from the target language as well as the source language phonemes. This new recogniser demonstrates moderate bilingual phoneme recognition capabilities. The bilingual recogniser is then further adapted to the target language using MAP and embedded Baum-Welch re-estimation techniques. This combination of adaptation techniques together with the data pooling strategy is uniquely applied in the field of cross-language recognition. The results obtained using this technique outperform all other techniques tested in terms of phoneme recognition rates, although it requires a considerably more time consuming training

process. It displays only slightly poorer phoneme recognition than the recognisers trained and tested on the same language database.

Keywords: acoustic distance measures, phoneme mapping, cross-language, transformation-based adaptation, MAP, MLLR, data pooling, embedded Baum-Welch re-estimation

Opsomming

Hierdie studie ondersoek ‘n unieke geoutomatiseerde benadering om een foneem af te beeld op ‘n ander, gebaseer op die akoestiese aftstande tussen die individuele foneme. Alhoewel die fokus van hierdie studie op kruis-taal toepassings is, kan die geoutomatiseerde benadering ook gebruik word vir toepassings op verskillende databasisse in dieselfde taal.

Die hoofdoel van hierdie navorsing is om die data van ‘n brontaal te gebruik, om die aanvanklike akoestiese modelle van die teikentaal af te rig wanneer min spraakdata beskikbaar is in die teikentaal. Om dit te doen, moet ‘n geoutomatiseerde tegniek gevind word om die twee datastelle se foneme op mekaar af te beeld. Deur dié tegniek toe te pas sal dit dan moontlik wees om die ontwikkeling van spraakherkenning stelsels vir nuwe tale te bespoedig. Die heidige navorsing wat op kruis-taal toepassings fokus gebruik ‘n fonetiese deskundige om die foneme op mekaar af te beeld. Hierdie foneem afbeelding navorsing het Engels-na-Afrikaans sowel as Afrikaans-na-Engels ondersoek. Dit is vooraf op hierdie tale toegepas, maar het ‘n fonetiese deskundige gebruik om die foneme op mekaar af te beeld.

Om die beste foneem afbeelding te bepaal word versillende akoestiese afstande met mekaar vergelyk. Die tipes afstandsmetings wat ondersoek is, is die Kullback-Leibler meting, die Bhattacharyya afstand, die Mahalanobis afstand, die Euclidiese afstand, die L2 afstand en die Jeffreys-Matusita afstand. Die afstand metings word getoets deur die kruis-databasis herkenningsresultate te vergelyk op die foneemmodelle wat geskep is van die TIMIT spraak databasis en ‘n plaaslike Suid-Afrikaanse SUN Spraakdatabasis. Deur die mees gepaste tipe afstandsmeting te kies, kan ‘n geoutomatiseerde proses om die foneme van die brontaal na die teikentaal af te beeld, gedoen word. Die beste afstandsmeting vir die afbeelding gee herkenningsresultate wat vergelykbaar is met die afbeelding wat deur ‘n fonetiese deskundige gedoen word.

Hierdie studie ondersoek ook die effek van die aantal Gaussiese mengsels op die afbeelding en die spraakherkenningstelsel se herkenning. Die resultate wys dat die herkenning asimptoties toeneem tot by ‘n limiet soos die aantal mengsels vermeerder. Verder het die studie ook die effek van die weglating van die delta en versnelling Mel frekwensie kepstrale koëffisiënte

ondersoek. Daar is gevind dat die insluiting van hierdie tydsfunksies die afbeelding en die herkenningstelsel se foneem herkenningstempo verbeter. Eksperimente is ook uitgevoer om die impak van die aantal Verskuilde Markov Modelle (VMM) te bepaal. Daar is bevind dat enkel toestand VMMe die optimale kruis-taal foneemherkenning resultate gee.

Nadat die afbeelding gedoen is, word spreker aanpassingstegnieke toegepas op die herkenners om die teikentaal se werksverrigting te verbeter. Die modelle van ‘n volledig afgerigte spraakherkenner in die brontaal word aangepas vir ‘n teikentaalmodel deur gebruik te maak van “Maximum Likelihood Linear Regression (MLLR)” gevolg deur “Maximum A Posteriori (MAP)” tegnieke. Ingebedde Baum-Welch herskattings (EBWR) word verder gebruik om die modelle aan te pas vir die teikentaal. Hierdie tegnieke lewer ‘n aansienlike verbetering in die foneemherkenningstempo. Alhoewel ‘n kombinasie van MLLR en MAP vooraf in spreker aanpassing gebruik is, is hierdie kombinasie van MLLR, MAP en EBWR aanpassingstegnieke ‘n unieke bydrae van dié studie.

Ten slotte word ‘n tegniek gebruik waar die data gepoel is om a nuwe herkenner te bou wat die outomatiese getransformeerde foneme van die teikentaal en die brontaal gebruik. Die nuwe herkenner demonstreer matige tweetalige foneemherkenningsvermoë. Die tweetalige herkenner word dan verder aangepas na die teikentaal deur MAP en ingebedde Baum-Welch herskattingstegnieke te gebruik. Hierdie kombinasie van spreker aanpassingstegnieke saam met die tegniek om data te poel is ‘n unieke tegniek wat in kruis-taal toepassings gebruik word. Die resultate wat hiermee verkry is, het beter as alle ander tegnieke wat getoets is presteer, alhoewel dit ‘n aansienlike meer tydrowende afrigtingsproses verg. Dit vertoon slegs effens slegter in terme van foneemherkenning as die herkenners wat op dieselfde databasis afgerig en getoets is.

Sleutelwoorde: akoestiese afstandsmetings, foneemafbeeldings, kruis-taal, transformasie-gebaseerde aanpassing, MAP, MLLR, gepoelde data, Baum-Welch herskatting.

ACKNOWLEDGEMENTS

It would be a travesty to pass off this dissertation without acknowledging the background contributions of so many people.

First and foremost, my sincere thanks and love go to my mum, my dad and my sister, Preetha, who have stood by me throughout, who have always believed in me and whose undying faith in me lifted me when I needed it the most. My thanks also go to the rest of my family and my friends who have supported and sacrificed with me throughout the past few years. I must make special mention of Karl Geggus, who set me on track to start the 'M'. My gratitude also goes to to Derik Thirion for his invaluable assistance during the intial stages of the experiments I conducted. I would also like to thank my Study Leader, Prof. Liesbeth Botha for the patience and guidance that she has shown me over the past years. To the numerous souls who have contributed in some way to this dissertation, be it through advice, through innovative ideas or by providing me with motivation or inspiration, I sincerely thank each and every one of you. I would be failing in my duty if I did not pay respect to my Maker for equipping me with the strength to withstand the "slings and arrows" that this Masters Degree threw my way.

I am reminded of one of my favourite speeches delivered by Anna Quindlen, a former Pulitzer Prize winner. In the address, she reminds us that there will be thousands of people in the world with the same qualifications as ourselves, and that what separates us from the rest of these people is our soul, our character. This sobering thought has helped me keep things in perspective.

Finally, to every person who has touched my life in some positive way, wherever you are, I dedicate this to you.

Contents

List of Abbreviations

ASR	Automatic Speech Recognition
Bha	Bhattacharyya
CDHMM	Continuous Density HMM
db	decibels
DCT	Discrete Cosine Transform
EBWR	Embedded Baum-Welch Re-estimation
EM	Expectation-Maximisation
Euc	Euclidean
FFT	Fast Fourier Transform
HMM	Hidden Markov Model
IPA	International Phonetic Association
JM	Jeffreys-Matusita
KL	Kullback-Leibler
LPC	Linear Predictive Coding
LVCSR	Large Vocabulary Continuous Speech Recognition
Mah	Mahalanobis
MAP	Maximum A Posteriori
MFCC	Mel-scaled Frequency Cepstral coefficients
ML	Maximum Likelihood
MLLR	Maximum Likelihood Linear Regression
msec	milliseconds
NN	neural network
pdf	probability density function
SD	Speaker Dependent
SI	Speaker Independent

List of Symbols

A	state transition probability matrix
D	feature vector dimension
M	the number of classes
N	the number of states in an HMM
O	observation sequence
T	the number of timeframes in an observation sequence
W	a transformation matrix
a	an HMM transition matrix
c	a mixture weight
q	a state sequence
$\sum$	a Gaussian covariance matrix
γ	the state occupancy variable
λ	the parameters of an HMM
μ	a Gaussian mean vector
σ	a Gaussian variance vector

Chapter 1

INTRODUCTION

Considerable time and effort has been vested in optimising the performance of large vocabulary continuous speech recognition (LVCSR) systems. However, this focus has traditionally been applied to a single language. While this may have been adequate previously, it no longer bears practicality if one considers how the global boundaries of the world are becoming integrated.

In speech technology systems, there is an ever-increasing interest in issues such as dialectal variation, the handling of foreign accents and cross-language applications. There is a need to extend the capabilities of existing, current and in-use speech recognition systems for which considerable time and effort has been expended to tune them to the same high-performing levels for a new language.

The questions then arise about how to adapt these existing speech recognition systems to a new language using the least amount of time and effort, and how to expedite the speech recognition process in these adapted recognisers to bring their performance up to an acceptable level.

Multilingual speech recognition systems have traditionally focused on four main areas:

- On building speech recognition systems that can accept speech input in different languages.

- On language identification applications in which the recognition system attempts to identify the language spoken by the individual.
- Studies have been conducted on sharing acoustic information between languages by constructing multilingual phone sets [37, 39, 40, 42].
- More recently, research has been conducted into using cross-language acoustic information with the goal of improving the performance of a recogniser in a new target language [3, 8].

When a recognition system has to be developed for a new language (either exclusively for the new language or for the new language in addition to existing languages) the recognition system optimised for another language can be adapted to the characteristics of the new language.

This dissertation presents a technique for building the initial acoustic phoneme models of a Hidden Markov Model (HMM) in a new (target) language using acoustic models trained in another (source) language. Very often, much less training data for the new language is available for building a completely new recogniser. By using the techniques proposed in this investigation, it would be possible to rapidly generate the initial acoustic models for the new language. Then, by utilising selected adaptation techniques [3, 8], these seed models can be refined by optimally using the data available in the target language. The adaptation strategies explored in this investigation include:

- adapting the models trained on a base language using target language data, and
- training models on multilingual (English and Afrikaans) pooled data, and then adapting the models using the target language data.

These techniques are relevant to any new language that has a high degree of overlap with the source language in terms of phonemes and are applied using both English and Afrikaans as the target languages.

Once the initial target language acoustic models have been created, acoustic model re-estimation techniques are used to fine-tune the target language phoneme models. This issue is

also explored in this investigation.

This study details the entire process to build a phoneme recogniser in a target language. It initially looks at how the automated process to map phonemes from one language to another can be applied to generate initial (seed) phoneme models in a target language, then the adaptation methodology suggested in [3] to build a new recogniser in a target language is applied, and finally refines the target language recogniser by re-estimating model parameters to closer match the available target language speech data.

1.1 Using intelligent techniques for phoneme recognition

More recently, there has been considerable focus on the use of intelligent techniques (such as neural networks) in speech recognition, particularly in phoneme recognition applications.

HMMs have traditionally been extensively used in speech recognition since they support both acoustic and temporal modelling. Acoustic variability covers different accents, pronunciations, pitches and volumes while temporal variability covers different speaking rates. Since HMMs are essentially a collection of states connected by transitions, they are ideally suited for modelling the temporal nature of speech. However, HMMs make a number of sub-optimal modelling assumptions [60]. One assumption is that all probabilities depend solely on the current state, not on the previous history. This is incorrect for speech applications. One consequence is that HMMs have difficulty modelling coarticulation, because acoustic distributions are strongly affected by recent state history. A further assumption is that there is no correlation between adjacent input frames. This is also false for speech applications. In accordance with this assumption, HMMs examine only one frame of speech at a time, and in so doing ignore the context of neighbouring frames. A further problem with using HMMs is that the HMM continuous density models suffer from model mismatch, i.e., there is a poor match between their a priori choice of statistical model (normally chosen to be a mixture of k Gaussians) and the true density of acoustic space.

Neural networks (NNs) have long since been used in pattern recognition applications. They have the ability to learn complex, non-linear functions, they have a high tolerance for noise and they generalise well, which is important since speech patterns are never exactly the same [60]. Time delay neural networks (TDNNs) [5] have previously been used in speech recognition applications. TDNNS are implemented with time delay connections, such that each input is subject to one or more time-delays. The major drawback of using NNs in speech recognition has previously stemmed from how they will be used for temporal modelling, since they usually perform best best in static or temporally localised pattern recognition applications.

A more recent approach has focussed on using NN-HMM hybrid models [60, 61, 62, 63], where NNs are used for acoustic modelling and HMMs for temporal modeling. The simplest way to integrate HMMs and NNs would be to implement various pieces of HMM systems using NNs. NNs are nonparametric models that do not suffer from quantisation error or make detailed assumptions. Furthermore, NNs can accommodate any size input window since the number of weights required in a network simply grows linearly with the number of inputs. Thus NNs lend themselves more naturally to context sensitivity than an HMM. However, in practice, the number of adjacent frames that can analysed simultaneously is limited. A further drawback of the hybrid model is that the assumption that all probabilities depend solely on the current state, independent of previous history affects hybrid speech recognisers as well since this is a property of the HMM temporal model.

A number of studies using hybrid phoneme recognisers have been carried out. Tebelskis [60] demonstrated the use of Linked Predictive Neural Networks (LPNNs) in phoneme classification, where each phoneme class was modeled by a separate neural network, and each network tried to predict the next frame of speech given some recent frames of speech. It was found that the hybrid recogniser had better acoustic modeling accuracy, better context sensitivity and more natural discrimination than using a standard HMM. Tebelskis obtained a 4.5% recognition improvement over standard HMMs.

Johansen [62] conducted a comparison of different model architectures for TIMIT phoneme recognition. A Guassian-based HMM was compared with two multilayer perceptron-HMM hybrids. Johansen found that there were very small differences between the phoneme recognition results of the HMM and hybrid architectures.

Torkkola [63] demonstrated using learning vector quantisation (LVQ)-based codebooks with HMMs in speech recognition. It was found that modeling classwise quantisation errors of LVQ by continuous-density hidden Markov models lead to significant improvements over the conventional HMM techniques. It was suggested that the resulting system could be used as a phonetic recognition engine in a large-vocabulary continuous-speech recognition system, but at the cost of increased computational complexity.

However, the main purpose of this particular investigation is to assess the viability of automatically mapping phonemes from a source language to a target language using acoustic distance measures. Although using intelligent techniques in phoneme recognition is a further possibility, this is beyond the scope of the present investigation and as such, only standard HMM techniques are considered in this study.

The next sub-section considers both the classical and modern approaches to speaker adaptation.

1.2 Speaker adaptation

Speaker adaptation techniques have traditionally been applied when moving from a speaker independent (SI) scenario to a speaker dependent (SD) one, i.e. the goal has centred around attempting to adapt the acoustic parameters of an existing recogniser to improve its performance for a new speaker. This adaptation usually involves minimal adaptation data, and the challenge is to optimise the use of this adaptation data in the most efficient manner possible.

A different application of speaker adaptation was demonstrated in [3]. Rather than applying speaker adaptation in the conventional manner, the study looked at adapting acoustic models of a source language using speech data from speakers in a target language. This technique was shown to be a viable option in that study. This implies that using similar techniques in the present study would also lead to favourable results.

1.3 Re-use of acoustic information for cross-language applications

A major stumbling block in the re-use of acoustic information for cross-language acoustic model training is the actual format of the speech data. In traditional speaker adaptation, there is a consistent format of the data that is used for adaptation since the problem involves the same language. Even if the adaptation speech data is from a different database, the phoneme mappings are usually one-to-one and consistent. For cross-language applications, however, problems are encountered since the different languages have different phoneme sets. This problem can be overcome if there is some reliable mechanism to automate the mapping process, either through the use of phonetic experts, or through the use of distance metrics. Mapping exercises done by phonetic experts are subjective, and are not always repeatable – there is no guarantee that the phonetic expert will map the same phoneme set in exactly the same manner on another occasion. Bearing in mind that it is not always easy to find someone with expert phonetic knowledge, this option is not always feasible. Finding an optimum distance metric for this mapping has not been extensively explored, especially within the cross-language realm. It is this problem that is addressed in the current study.

Kohler [7] investigated methods to develop multilingual phone models for a telephone-based speech recognition system built for six languages (French, German, Italian, Portuguese, Spanish and American English). One of the methods that he considered was a direct mapping to the phone set of the International Phonetic Association (IPA). This method did not use any language-dependent acoustic information but relied on the accuracy and the consistency of the phonetic transcriptions. This means that a particular phoneme unit may be mapped to a

phoneme set that is orthographically closer, rather than acoustically similar. He was able to reduce the phone set from 232 language-dependent models to 47 multilingual models (where "multilingual" implies that the phone can be found in more than one language) and 48 monolingual models. Although this approach reduced the computational effort required for the multilingual recognition task, the study noted a 5% decrease in recognition rate. The study also investigated a multilingual phone clustering technique where a log-likelihood-based distance measure was used to compute the similarity between two phone models. This technique yielded better results than the direct IPA-mapping one, with a decrease of 1.8% when compared to language-dependent recognition.

Mukherjee *et al.* [48] proposed a method for determining the initial phoneme models for a new language (Hindi) using an already trained system in a base language (English). The acoustic similarity between the phone models of the two languages is determined in the Linear Discriminant Analysis (LDA) space. Using this technique, they obtained a phoneme classification rate of 26.99% without performing any additional adaptation.

This study shows that by selecting an appropriate distance measure, an automated procedure to map phonemes from the source language to the target language can be applied, with phoneme recognition results comparable to a manual mapping process undertaken by a phonetic expert.

The author conducted a number of experiments on the cross-language phoneme recognisers to investigate the effect of certain parameter variations. The results of this investigation indicate that the recogniser's performance increases up to a limit as the number of mixture components increase. The effect of excluding the Mel Frequency delta and acceleration cepstral coefficients was also explored. It is found that the inclusion of these temporal features help improve the recognition system's phoneme recognition rate. Experiments are also carried out to determine the impact of the number of HMM recogniser states. It is found that single-state HMMs deliver the optimum cross-language phoneme recognition results.

Speaker adaptation strategies are then applied on the recognisers to improve their target-language performance. The models of a fully trained speech recogniser in a source language

are adapted to target-language models using the MLLR followed by the MAP techniques. Embedded Baum-Welch re-estimation is then used to further adapt the models to the target language. These techniques show a considerable improvement in the phoneme recognition rate.

Finally, as a separate experiment, a data pooling technique is applied to build a new recogniser using the automatically mapped phonemes from the target language as well as the source language phonemes. This new recogniser demonstrates moderate "bilingual" phoneme recognition capabilities. The "bilingual" recogniser is then further adapted to the target language using MAP and embedded Baum-Welch re-estimation techniques. The results obtained using this technique outperform all other techniques, although it is a considerably more time consuming training process.

1.4 Organisation of dissertation

The organisation of this dissertation is as follows. The background theory underlying Automatic Speech Recognition (ASR) systems is discussed in Chapter 2 with emphasis on hidden Markov models (HMMs) and speaker adaptation techniques. The theory of the techniques that are used in this study is also presented in Chapter 2. Chapter 3 discusses the distance measures that are used in the investigation.

The experimental methodology applied is explained in Chapter 4. The experiments, the experimental results and a discussion of the findings are described in Chapters 5, 6 and 7. Chapter 5 describes the data mapping experiments. The MLLR and MAP adaptation experiments are included in Chapter 6. The data pooling and embedded Baum Welch Re-estimation experiments and findings are discussed in Chapter 7. Finally, the conclusion is presented in Chapter 8. Suggestions on future related research are also included.

An overview of the software used in this investigation is given in Appendix A. The TIMIT and SUN Speech databases that were used in the experiments are described in more detail in Appendix B. The SUN Speech to TIMIT as well as the TIMIT to SUN Speech phonemic

mappings are listed in Appendices C and D respectively.

1.5 Contributions of dissertation

The contributions of the research documented in this dissertation include the following points:

- The author presents a new strategy for the automatic mapping of phonemes from a source language to a target language. This is of particular interest in cross-language speech recognition applications. Previous research has focused either on manually mapping phonemes, or has utilised techniques other than the distance-based one used in this study. The performance of several distance measures is then evaluated to quantify the acoustic distance between phonemes, and show that selected distance measures consistently outperform the others. It is also shown that the best performing distance measures are able to match the manual phoneme mapping procedures carried out by a phonetic expert.
- The new automated strategies proposed in this dissertation are applied to English-to-Afrikaans phoneme mapping, as well as Afrikaans-to-English phoneme mapping. This has been previously applied to these language instances, but utilising manual phoneme mapping methods.
- This investigation applies two of the speaker adaptation strategies suggested in [3, 8] for cross-language applications. The mean and variance of the HMM models of a fully trained speech recogniser in a source language are adapted to models for a target language using the MLLR followed by the MAP techniques. In addition, embedded Baum-Welch re-estimation (EBWR) is applied to further adapt the models to the target language. Although a combination of MLLR and MAP techniques have been used previously in speech adaptation studies, the combination of MLLR, MAP and EBWR in cross-language speech recognition is a unique contribution of this study. It is shown that these techniques allow rapid development of a speech recogniser in the new target language.
- The data pooling strategy suggested in [3, 8] is applied to build a new recogniser using

the automatically mapped phonemes from the target language and the source language phonemes. Further to the current research in this field, the author conducts experiments to assess the bilingual speech recognition performance of the recogniser and show moderate success with its bilingual capabilities. MAP adaptation techniques (to both the mean and variance) and embedded Baum-Welch re-estimation are applied to further adapt this "bilingual" recogniser to the target language. This combination of adaptation techniques together with the data pooling strategy is uniquely applied in the field of cross-language recognition. These results are compared to those obtained using the MLLR-MAP technique and show the data pooling followed by adaptation technique outperforms the other techniques tested, although it is a considerably more time consuming process.

1.6 Publications

The following two publications are further contributions of this investigation:

- J.J. Sooful and E.C. Botha, "An acoustic distance measure for automatic cross-language phoneme mapping*," Proceedings of the Twelfth Annual Symposium of the Pattern Recognition Association of South Africa*, Franschhoek, South Africa, pp. 99-102, 29-30 November 2001.
- J.J. Sooful and E.C. Botha, "Comparison of acoustic distance measures for automatic cross-language phoneme mapping," *Proceedings of the 10th International Conference on Spoken Language Processing*, Denver, Colorado, pp. 521-524,September 2002.

Chapter 2

BACKGROUND THEORY

This chapter discusses the background theory that underpins the research that was conducted in this dissertation. Firstly, an overview is given of the speech sounds and how they are produced. Next the relevant terminology and basic algorithms and techniques relating to speech recognition are discussed. This includes a description of pre-processing and feature extraction techniques. A detailed discussion of the theory pertaining to Hidden Markov Models then follows. Next multilingual speech processing is covered, focusing on speaker adaptation theory and techniques. Two well-known techniques, Maximum Likelihood Linear Regression (MLLR) and Maximum A Posteriori (MAP), are discussed in detail.

2.1. Speech sounds

Automatic speech recognition (ASR) systems generally assume that a speech signal is a sequence of one or more basic units. These basic units could be phones, syllables or words.

For the purposes of this investigation, it is imperative to draw a distinction between “phones” and “phonemes”. *Phones* refer to actual sounds produced by the vocal tract while speaking. *Phonemes* are defined as the smallest linguistic unit with meaning. In essence, this implies that the phonemes of a language comprise a theoretical set of units that are sufficient to convey all the meaning in the language. Due to a variety of factors (such as accent, dialect and

physiology), a phoneme will have various acoustic manifestations. The related and actual sounds that are produced when speaking are called phones.

More recently, some interest has also been directed at *graphemes*, and indeed grapheme-based speech recognition systems [38]. Graphemes refer to the letters and letter combinations that represent a phoneme, for example *f, ph,* and *gh* for the phoneme [f]. Grapheme-based speech recognisers are built by using the grapheme labels (rather than the phoneme labels) when training.

The speech production mechanism in humans whereby air is expelled from the lungs and forced along the trachea and through the glottis can be controlled in different ways to produce voiced, unvoiced and plosive sounds [5]. These and other relevant speech terminology will now be explained.

Voiced sounds such as "oh" or "aah" are produced when the vocal cords are tensed together and vibrate as the air pressure builds up, forcing the glottis open, and then subsides as the air passes through. The vibration that is produced has a frequency spectrum rich in harmonics at multiples of the fundamental frequency (*pitch*).

Unlike voiced sounds, *unvoiced* sounds do not cause the vocal cords to vibrate. These sounds may be fricative or aspirated. Fricative sounds, such as "sh", "s" or "f", are generated at some point in the vocal tract. As air is forced past it, turbulence occurs causing a random noise. Since the points of constriction tend to occur near the front of the mouth, the resonances of the vocal tract have little effect on the sound being produced. In aspirated sounds, such as "h" in "**h**ay", turbulent airflow occurs at the glottis, as the vocal cords are held slightly apart. Resonances of the vocal tract modulate the spectrum of the random noise, as heard in whispered speech.

Plosive sounds, such as the "puh" sound at the beginning of the word "**p**ea" or the "duh" sound at the beginning of "**d**ay", are created when the vocal tract is closed at some point, allowing air pressure to build up, before it is suddenly released. This may occur with or without vocal cord vibration. The presence or absence of vocal-cord vibration distinguishes

the voiced stops (as in "**b**ad", "**g**one") from the unvoiced stops ("**k**ill", "**t**on"). Plosives are characterised by transient bursts of energy. As a result, their properties are highly influenced by the sounds that precede or follow them.

Formants are resonances produced by the tubular shape of the vocal tract. The vocal tract may assume many different shapes giving rise to different resonant or formant frequency values. In continuous speech, formant frequencies are constantly changing.

Vowels are produced when sound radiates from the mouth with no nasal coupling. The tongue shape remains fairly fixed and each vowel is characterised by the forward/backward and raised/lowered positions of the tongue. Vowels may be classified as *front*, as in the words "b**i**t", "s**a**t" or "r**e**d". Examples of *middle* vowels are found in the words "b**ir**d", "r**u**t" or "th**e**". Examples of *back* vowels appear in the words " cr**u**de", "b**ore**d" and "w**oo**d". Each vowel is characterised by the values of the first three or four resonances (formants) of the vocal tract. Semivowels consist of glides and liquids. Glides (as in "**w**ent" and "**r**an") are dynamic sounds except that articulators move much more rapidly from one static vowel position to another. Liquids (as in "**y**ou" and "**l**et") are static gestures with the oral tract partially closed at some point.

Diphthongs are a combination of two vowel sounds. They are similar to vowels with the main difference being that the gesture is created when the articulators move slowly from one static vowel position to another. Examples include "b**oy**" and "b**ai**t."

Nasal sounds are produced by vocal-cord excitation with the vocal tract totally constricted at some point along the oral passageway such that sound is radiated from the nostrils. Examples of nasals are "**m**om", "si**n**g" and butto**n**."

Fricatives are produced when turbulent airflow occurs at a point of constriction in the vocal tract. The point of constriction occurs near the front of the mouth and its exact location characterises the particular fricative sound produced. Sound is radiated from the lips via the front cavity. Unvoiced fricatives, as in "**s**at", "**th**in" and "**f**it", are produced without vocal-cord vibration. Voiced fricatives, to be found in '**v**ision", "**th**en" and "**z**one", are produced when

the vocal cords are vibrating.

Affricates are either voiced as in the word "**j**u**dg**e" or unvoiced as in the word "**ch**ur**ch**." These sounds are produced when a stop and fricative consonant are both shortened and combined.

The basic terminology and techniques associated with the processing of these sounds into a form that can be used in automatic speech recognition applications follow.

2.2. Basic speech recognition terminology

The discussion that follows is based on the speech signal being transformed from a time signal in temporal space into multi-dimensional feature space.

Let $\boldsymbol{O}$ represent the sequence of acoustic feature vectors that has been observed, such that O is defined by:

$$O = o_1, o_2, \ldots, o_T \tag{2.1}$$

where $\boldsymbol{o}_T$ is the feature vector observed at time T [5].

Let:

$$W = w_1, w_2, \ldots, w_N \tag{2.2}$$

represent a sequence of N basic units that are to be recognised.

The speech recognition task then translates into computing the posterior probability:

$$P(W/\mathbf{O}). \tag{2.3}$$

Given the dimensionality of the observation sequence $\boldsymbol{O}$, there is no way to compute this probability directly. However, using Bayes' Rule [5, 18], it is possible to calculate the probability that a given sequence of basic units can generate a set of acoustic vectors. Using Bayes' Rule, Equation (2.3) reduces to:

$$P(W/\mathbf{O}) = \frac{P(\mathbf{O}/W).P(W)}{P(\mathbf{O})} \quad (2.4)$$

where *P(W)* is the prior probability of observing a sequence of basic units, *W, P(**O**)* is the prior probability of a set of acoustic observation vectors, ***O***, and *P(**O**/W)* is the probability that a set of acoustic vectors, ***O***, will be observed when the sequence of basic units, *W*, is produced.

P(W) is estimated if there is sufficient speech data for the recognition task. Usually, *P(W)* is language-specific and is determined by a language model in the form of a *bi-gram* or *tri-gram*. A bi-gram model computes the probability of finding a specific speech unit given the preceding speech unit. It can be obtained by statistically analysing a large text corpus. Similarly, a tri-gram gives the probability of finding a specific speech unit given the preceding two speech units.

The statistical parameters relating to *P(**O**/W)* can be reliably estimated, provided that there is a sufficient amount of representative training data. Speech recognition systems normally search for the sequence of basic units, *W*, that maximizes *P(**O**/W)*.

The prior probability *P(**O**)* of a set of speech observation vectors, ***O***, is often assumed to be constant, and is omitted from Equation (2.4).

2.2.1. Pre-processing

A technique that is often used to spectrally flatten the speech signal is *pre-emphasis*. The need for pre-emphasis stems from the speech production model of voiced speech. This model shows that there is a –6 dB/octave decay in speech radiated from the lips as frequency increases. This is a combination of a -12 dB/octave trend due to the voiced excitation source and +6 dB/octave trend due to radiation from the mouth. This essentially means that, for each doubling in frequency, the signal amplitude, and hence the measured vocal tract response, is reduced by a factor of 16. It is therefore desirable to compensate for the -6 dB/octave roll-off by pre-processing the speech signal to give a +6 dB/octave lift in the appropriate range so that the measured spectrum has a similar dynamic range across the entire frequency band. It is unnecessary to apply pre-emphasis in the case of unvoiced speech since there is no spectral

trend to remove. However, pre-emphasis for unvoiced speech is included for simplicity of implementation. In the time domain, the pre-emphasis high pass filters that are implemented take the form:

$$x_i = x_i - a\, x_{i-1} \tag{2.5}$$

if **x** is an *N*-sample digitised signal, and $0.9 \leq a \leq 1.0$.

To extract the short-time features of the pre-emphasised, normalised speech signal, it is *blocked* into short segments called *frames*. The frame size used in speech recognition applications normally varies from 10 msec to 30 msec, and the speech contained within each frame is assumed to be stationary.

If the speech signal is blocked into *Z* frames, the step size, *V*, is usually chosen between 20% and 50% of the frame size *Z* [17]. This means that successive frames will overlap by *Z –V* samples. This overlapping means that the resulting spectral features are correlated from frame to frame, resulting in a smoother feature set. If *M* frames cover the entire signal, the data pertaining to the speech signal may be stored in an *M* x *Z* matrix, **Y**, with each row $\boldsymbol{y}_i$ representing the frames. **Y** would then be represented as:

$$y_{ij} = x_{v_{i+j}} \qquad j = 0, 1, \ldots, Z\text{-}1,\ i = 0, 1, \ldots, M\text{-}1. \tag{2.6}$$

Each individual frame is *windowed* to minimise the signal discontinuities in the time domain at the edges of each frame caused by the blocking. The smoothing window ensures that the signal value at the beginning and end of each frame slowly reduces to zero. One of the most common windowing functions used in speech processing is the Hamming window. This is described in Equation (2.7):

$$z_j = 0.54 - 0.46\cos\left(\frac{2\pi j}{Z-1}\right), \qquad j = 0, 1, \ldots, Z\text{-}1. \tag{2.7}$$

Multiplying the data stored in matrix **Y** by the Hamming window function gives:

$$y_{ij} \Leftarrow y_{ij} z_j, \qquad j = 0, 1, \ldots, Z\text{-}1,\ i = 0, 1, \ldots, M\text{-}1. \tag{2.8}$$

A fast Fourier transform (FFT) is performed on each windowed signal. The Fourier transform provides a mathematical basis for determining the frequency spectrum of a continuous time-domain signal. If the Fourier transform of $\mathbf{y}_i$ is represented by $F\mathbf{y}_i$, then the power spectrum of each window is obtained by squaring each Fourier coefficient to obtain its real value, which leads to:

$$s_{ij} = |(Fy_{ij})|^2, \qquad j = 0, 1, \ldots, Z-1, \quad i = 0, 1, \ldots, M-1. \tag{2.9}$$

2.2.2. Feature extraction – Mel-scaled cepstral coefficients

As with any robust speech recognition (and indeed pattern recognition) system, feature extraction needs to be performed to reduce the amount of speech data into a manageable amount of information without discarding valuable information.

Empirical evidence has shown that speech recognition systems that mimic the non-linear human perception of sound exhibit better recognition performance than those that do not [5, 18]. The mel frequency scale is widely used to take into account the subjectivity of the human ear. The mel scale represents the perceptual relationship between pitch (in mels) to frequency (in Hertz).

Linear predictive coding (LPC) analysis is a feature extraction technique that is sometimes used. However, *mel-scaled filter bank* analysis has repeatedly shown to outperform LPC analysis techniques [5, 7, 10]. This consists of a number of overlapping triangular filters that have linearly spaced frequencies and fixed bandwidth on the mel scale, as is evident in Figure 2.1 below:

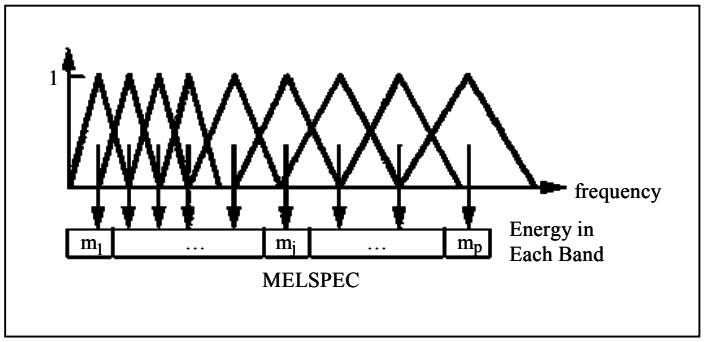

Figure 2.1: Mel-scale filter bank

Each filter has a triangular frequency response that is equal to unity at its centre frequency, and decreases linearly to zero at the center frequency of the two adjacent filters. Normally, the triangular filters are spread over the whole frequency from zero to the Nyquist frequency. However, band limiting is used to constrain the allocation of filters to frequency regions where there is meaningful information.

If f_{min} and f_{max} (equal to Nyquist frequency) represent the lowest and highest frequencies of the meaningful frequency band over which the filter bank extends, f_c is the center frequency of a filter, f_l and f_h are the center frequencies of two adjacent filters and let

$$\eta = \frac{Z-1}{f_{\max} - f_{\min}}, \tag{2.10}$$

$$I_r = \eta\,(f_h - f_{min}),$$

then

$$f_{ij} = \begin{cases} \dfrac{1}{f_c - f_l}\left(\dfrac{j}{\eta} + f_{\min} - f_l\right) & \text{if} \quad I_l \le j \le I_c \\ 1 + \dfrac{1}{f_c - f_h}\left(\dfrac{j}{\eta} + f_{\min} - f_c\right) & \text{if} \quad I_c \le j \le I_r \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

Normally, the number of filters in the filter bank, K, varies between 13 and 24.

To implement this filter bank, the window of speech data is transformed using a Fourier transform and the magnitude is taken. Each FFT magnitude coefficient is multiplied by the corresponding filter gain and the results accumulated. This yields a weighted sum representing the spectral magnitude in that filter bank channel.

The log of each mel-spaced filtered coefficient is then taken. A discrete cosine transform (DCT) is performed on the coefficients to yield the mel-scaled cepstral coefficients or MFCCs. These are calculated from the log filter bank amplitudes $\{m_j\}$ and are represented by:

$$c_i = \sqrt{\frac{2}{K}} \sum_{j=1}^{K} m_j \cos\left(\frac{\pi i}{K}(j-0.5)\right) \quad j = 0, 1, \ldots, Z\text{-}1, \; i = 0, 1, \ldots, M\text{-}1. \qquad (2.12)$$

The performance of speech recognition systems can be significantly enhanced by adding time derivatives to the basic static parameters [5, 9, 26]. The addition of these temporal aspects into the feature vector broadens the scope of the frames. Since the data contained in the cepstral log vector is discrete, its first and second order derivatives can be calculated from an orthogonal polynomial approximation of the data trajectory. The *delta* (first derivative) and *delta-delta* or *acceleration* (second derivative) coefficients may be included in the acoustic feature vector.

One problem associated with cepstral coefficients is that the higher order cepstra are quite small, leading to a high variance when going from low to high order cepstra. This can be overcome by re-scaling or *liftering* the cepstral coefficients using some value, L, according to the formula

$$c_n' = \left(1 + \frac{L}{2}\sin\frac{\pi n}{L}\right) c_n . \qquad (2.13)$$

The energy of an utterance contains important information about the phonetic identity of the utterance [5], whether it represents speech, silence or noise. The energy can also be used to distinguish between classes of sounds. Vowels, for example, have a higher energy than fricatives. The energy is calculated as a log of the signal energy. For speech samples s_n, the

computed energy is

$$E = \log \sum_{n=1}^{N} s_n^2 , \qquad i=1, 2, \ldots, N. \tag{2.14}$$

The delta and delta-delta coefficients can be computed for the energy component as well. If the energy component is expressed in terms of power, then its derivative gives an indication of the changes in the amplitude of the speech signal.

Energy normalisation is often included as part of the pre-processing of the speech signal. This forces the maximum signal amplitude to 1. If there are N speech signals to be normalised, then energy normalisation is implemented by the following equation:

$$Enormalised_i = E_i - E_{\text{maximum}} + 1 \qquad i = 1, 2 \ldots N. \tag{2.15}$$

Now that the basic terminology and techniques pertaining to speech signals have been discussed, the fundamental concepts of Hidden Markov Models can be discussed in more detail.

2.3. Hidden Markov models

Continuous Density Hidden Markov Models (CDHMMs or often referred to as just HMMs) represent a statistical method to model the spectral properties of the frames of a pattern. When HMMs are used to model speech, an assumption is made that the speech signal can be described as a parametric random process and that the parameters describing the process can be estimated precisely. A requirement for such a system is that it is stationary, meaning that its statistical properties should remain constant over time.

Speech is not a stationary process; it is assumed to be piecewise stationary, over short intervals from one sample to the next. This assumption means that statistically meaningful parameters can be estimated from the acoustic features of a speech signal if these parameters are extracted at regular, adequately small intervals.

The discussion that immediately follows deals with the theory behind HMMs. A more comprehensive discussion on HMMs can be found in [5], although the salient aspects relating to HMMs are covered here. HMMs are defined in terms of states, transitions and other basic units. Next the effectiveness of HMMs is covered in terms of the 3 basic HMM problems and their solutions. Parameter estimation algorithms and statistical language models are introduced as part of this discussion.

An HMM is denoted by λ and is described by two sets of parameters [3, 5]:

- a state transition matrix, $A = \{a_{ij}\}$. Each a_{ij} denotes the discrete probability of making transitions from a state i to a state j. Further constraints on the state transition probabilities are:

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \text{, and} \tag{2.16}$$

$$a_{ij} \neq 0 \qquad \text{only for } j = i \text{ or } j = i+1. \tag{2.17}$$

- a continuous state probability density function $b_j(\mathbf{o}_t)$ reflecting the likelihood of observing observation vector $\mathbf{o}_t$ in state j.

An HMM is a finite state machine that changes state once every time unit. Each time t that a state j is entered, a speech observation vector $\mathbf{o}_t$ is generated from the probability density $b_j(\mathbf{o}_t)$.

Usually, only first order HMMs are considered since each transition probability to the next state depends only on the current state, and not the previous states. The standard approach in speech recognition is to use standard left-to-right HMMs without skipping transitions. The left-to-right modelling assumes that the observation sequences corresponding to the same HMM traverse the same discrete sequence of statistical properties in much the same way as a speech signal.

The joint probability that the observations sequence **O** is generated by the model λ moving through the state sequence X is calculated as the product of the transition probabilities and the

output probabilities. However, in practice, only the observation sequence **O** is known and the underlying state sequence X is hidden, hence the Markov model is known as *hidden*. The states are thus not directly observed, but through modelling of the observation distributions in each state.

Gaussian mixtures are used to model the observation probability density functions. In order to use such a model, the assumption is made that the frames of the speech utterance are independent of each other. The advantages of using a Gaussian mixture model as the likelihood function are that it is computationally inexpensive, it is based on a well-understood statistical model, and, for text-independent tasks, is insensitive to the temporal aspects of the speech, modeling only the underlying distribution of acoustic observations from a speaker [23]. The latter temporal insensitivity of Gaussian models is also a disadvantage in that higher levels of information about the speaker conveyed in the temporal speech signal are not used.

For an observation $\mathbf{o}_t$ at time t in state j, this density function is of the form:

$$\begin{aligned} b_j(\mathbf{o}_t) &= \sum_{k=1}^{K} c_{jk}\mathbf{N}\left[o_t, \mu_{jk}, \Sigma_{jk}\right] \\ &= \sum_{k=1}^{K} c_{jk}(2\pi)^{-D/2}\left|\Sigma_{jk}\right|^{-1/2} e^{-\left(\frac{1}{2}\right)(\mu_{jk}-\mathbf{o}_t)^T \Sigma_{jk}^{-1}(\mu_{jk}-\mathbf{o}_t)}. \end{aligned} \tag{2.18}$$

In Equation (2.18),
K is the number of mixture components,
D is the number of feature vector elements,
c_{jk} is the weight associated with the k^{th} mixture in the j^{th} state,
N is the multivariate normal density,
μ_{jk} is the mean of the k^{th} mixture in the j^{th} state, and
Σ_{jk} is the covariance matrix of the k^{th} mixture in the j^{th} state.

A simplifying assumption that reduces the number of parameters (and hence the computation effort) in the aforementioned equation is that Σ_{jk} is a diagonal matrix. This assumption is made since the elements of $\mathbf{o}_t$ are largely uncorrelated. This reduces the probability density function to:

$$b_j(\mathbf{o}_t) = \sum_{k=1}^{K} c_{jk} \prod_{l=1}^{D} (2\pi)^{-1/2} \sigma_{jkl}^{-1} e^{-(o_{tl}-\mu_{jkl})^2 / 2\sigma_{jkl}^2}, \quad (2.19)$$

where $\mathbf{o}_{tl}$ is the l^{th} element of the observation vector at time t, μ_{jkl} is the l^{th} element of the mean vector in mixture k of state j, and σ^2_{jkl} is the l^{th} variance value on the diagonal of the covariance matrix Σ_{jk}.

For the discussion on HMMs that follows, the notation that will be used is defined below:

N = number of states in the model, with individual states being numbered as $\{1,2,\ldots,N\}$, and with q_t referring to the state at time t,

M = total number of distinct observations per state, the elements of the observation set are denoted as $V = \{v_1, v_2, \ldots, v_M\}$,

$\boldsymbol{O}_t$ will denote the observation symbol observed at instant t,

$A = \{a_{ij}\}$ is the state transition probability matrix, a_{ij} refers to the probability of a transition from state i to state j. The elements of A are given by

$$a_{ij} = P\,[q_{t+1} = j | q_t = i], \qquad 1 \le i \le N, \quad (2.20)$$

$B = \{b_j(k)\}$ is the observation symbol probability matrix, $b_j(k)$ defines the symbol distribution in state j, $j=1,2,\ldots,N$. The elements of B are given by

$$b_j(k) = P\,[o_t = v_k | q_t = j], \qquad 1 \le k \le M, \quad (2.21)$$

$\pi = \{\pi_i\}$, the initial state distribution vector, with elements

$$\pi_i = P[q_1 = i], \qquad 1 \le i \le N, \quad (2.22)$$

$\lambda = (A, B, \pi)$ denotes the parameter set for a given HMM.

An example of a classical left-to-right HMM with 3 states (N=3) is shown in Figure 2.2.

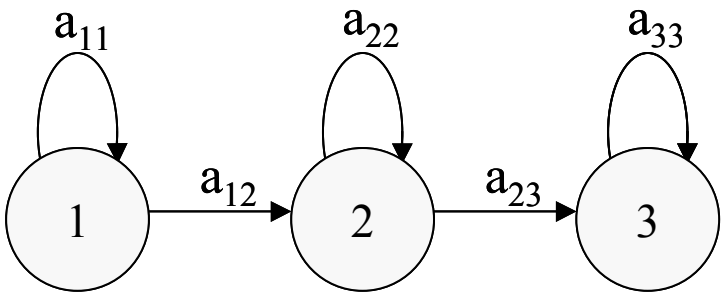

Figure 2.2: A left-to-right HMM with three states

Most applications of HMMs are reduced to solving three issues:

Problem 1: Given the model $\lambda = (A, B, \pi)$, how to compute $P(\mathbf{O}/\lambda)$, the probability of the occurrence of the observation sequence $\mathbf{O} = (o_1, o_2, \ldots, o_T)$.

Problem 2: How to find the most likely state sequence $\mathbf{q} = (q_1, q_2, \ldots, q_T)$ given the model $\lambda = (A, B, \pi)$ and an observation sequence $\mathbf{O} = (o_1, o_2, \ldots, o_T)$.

Problem 3: How to adjust the HMM parameter set $\lambda = (A, B, \pi)$ such that $P(\mathbf{O}/\lambda)$ is maximized.

2.3.1. Problem 1 – Probability evaluation

One way to determine $P(\mathbf{O}/\lambda)$ is to find $P(\mathbf{O}/\mathbf{q}, \lambda)$ for a fixed state sequence $\mathbf{q} = (q_1, q_2, \ldots, q_T)$ and then multiply it by $P(\mathbf{q}/\lambda)$ and then to find the accumulated sum over all q's. So,

$$P(\mathbf{O}/\mathbf{q}, \lambda) = b_{q1}(\boldsymbol{o}_1) b_{q2}(\boldsymbol{o}_2) \ldots b_{qT}(\boldsymbol{o}_T) \quad \text{and} \tag{2.23}$$

$$P(\mathbf{q}/\lambda) = \pi_{q1} a_{q1q2} a_{q2q3} \ldots a_{qT\text{-}1qT}\,, \tag{2.24}$$

which leads to

$$\begin{aligned} P(O/\lambda) &= \sum_{\text{all q}} P(\mathbf{O}/\mathbf{q}, \lambda) P(\mathbf{q}/\lambda) \\ &= \sum_{q_1, q_2, \ldots q_T} \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1 q_2} b_{q_2}(\mathbf{o}_2) \ldots a_{q_{T-1} q_T} b_{q_T}(\mathbf{o}_T) \end{aligned} \tag{2.25}$$

From Equation (2.26), it can be seen that the summand of this equation requires $2T$-1 multiplications and there are N^T distinct possible state sequences $\mathbf{q}$. Thus to calculate Equation

(2.26) directly will require $2TN^T$ multiplications. A more efficient technique to solve Problem 1 is the forward-backward procedure [5, 19].

2.3.2. Forward procedure

Consider the variable $\alpha_t(i)$ which is defined as:

$$\alpha_t(i) = P(\boldsymbol{o}_1,\boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, q_t = i/\lambda) \tag{2.26}$$

which expresses the probability of the partial observation sequence $\boldsymbol{o}_1,\boldsymbol{o}_2, \ldots, \boldsymbol{o}_t$ and state i at time t given the model λ. $\alpha_t(i)$ can be computed inductively as follows:

- Step 1

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1), \qquad 1 \le i \le N \tag{2.27}$$

- Step 2

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1}), \qquad \begin{matrix} 1 \le t \le T-1 \\ 1 \le j \le N \end{matrix} \tag{2.28}$$

- Step 3

$$P(\mathbf{O}/\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.29}$$

In the first step, the forward probabilities are initialised to the value of the joint probability of state i and initial state $\boldsymbol{o}_1$.

In the second step, the probability of the partial observation sequence up to time t+1 and state j at time t+1 needs to be computed. State j can be reached at time t+1 from the N possible states at time t. However, $\alpha_t(i)$ is the probability of the joint event that $\boldsymbol{o}_1,\boldsymbol{o}_2, \ldots, \boldsymbol{o}_t$ are observed and that the state at time t is i. Therefore, the product $\alpha_t(i)a_{ij}$ represents the joint event that $\boldsymbol{o}_1,\boldsymbol{o}_2, \ldots, \boldsymbol{o}_t$ are observed and state j is reached at time t+1 via state i at time t.

With j known, $\alpha_{t+1}(j)$ can be determined by multiplying the summed quantity by the probability of observing $\boldsymbol{o}_{t+1}$ in state j, $b_j(\boldsymbol{o}_t+1)$. The second step is repeated for all states j for a given time t and then iterated for $t = 1,2, \ldots, T$-1.

In the final step, the value of $P(\mathbf{O}/\lambda)$ is calculated as the sum of all the forward variables $\alpha_T(i)$

since

$$\alpha_T(i) = P(\boldsymbol{o}_1,\boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, q_T = i/\lambda). \quad (2.30)$$

The forward procedure requires N^2T calculations, much more favourable than the $2TN^T$ multiplications required for the direct computation.

2.3.3. Backward procedure

In a similar manner to the forward procedure, the backward variable $\beta_t(i)$ can be defined as:

$$\beta_t(i) = P(\boldsymbol{o}_{t+1},\boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T, q_t = i/\lambda). \quad (2.31)$$

The backward procedure considers the probability of the partial observation sequence from time t+1 to the end of the sequence, given the state i at time t and the model λ. $\beta_t(i)$ can be solved iteratively through the following steps:

- Step 1

$$\beta_T(i) = 1, \qquad 1 \le i \le N. \quad (2.32)$$

- Step 2

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j), \qquad t = T-1, T-2, \ldots, 1 \quad 1 \le i \le N. \quad (2.33)$$

- Step 3

$$P(\mathbf{O}/\lambda) = \sum_{i=1}^{N} \pi_i b_i(\mathbf{o}_1) \beta_1(i), \qquad 1 \le i \le N. \quad (2.34)$$

During the first step, $\beta_t(i)$ is arbitrarily set to 1 for all i. In order to have been in state i at time t and at the same time account for the observation sequence from t+1 onwards, all possible states j at time t+1 must be considered. The a_{ij} term accounts for the transition from state i to state j, while the $b_j(\boldsymbol{o}_{t+1})$ term accounts for the observation $\boldsymbol{o}_{t+1}$ in state j. The remaining part of the partial observation sequence from state j is accounted for by the $\beta_{t+1}(j)$ expression.

The backward procedure also has a computational complexity of N^2T calculations.

2.3.4. Problem 2 – Optimal state sequence

This problem deals with finding the optimal state sequence associated with any given observation sequence. This problem can be solved in a variety of different ways.

Consider the expression, $P(\mathbf{O}, \mathbf{q}/\lambda)$. From Equations (2.24) and (2.25), this can be expressed as:

$$\begin{aligned} P(\mathbf{O},\mathbf{q}/\lambda) &= P(\mathbf{O}/\mathbf{q},\lambda)P(\mathbf{q}/\lambda) \\ &= \pi_{i_1} b_{i_1}(\mathbf{o}_1) a_{i_1 i_2} b_{i_2}(\mathbf{o}_2) \ldots a_{i_{T-1} i_T} b_{i_T}(\mathbf{o}_T). \end{aligned} \tag{2.35}$$

Now define

$$\mathbf{U}(q_1, q_2, \ldots, q_T) = -\left[\ln(\pi_{i_1} b_{i_1}(\mathbf{o}_1)) + \sum_{t=2}^{T} \ln(a_{i_{t-1} i_t} b_{i_t}(\mathbf{o}_t)) \right], \tag{2.36}$$

then it can be seen that

$$P(\mathbf{O},\mathbf{q}/\lambda) = \exp(-\mathbf{U}(q_1, q_2, \ldots, q_T)). \tag{2.37}$$

This means that the problem of optimal state estimation ,

$$\max_{\{q_t\}_{t=1}^{T}} P(O, q_1, q_2, \ldots q_T/\lambda) \tag{2.38}$$

becomes equivalent to

$$\min_{\{q_t\}_{t=1}^{T}} \mathbf{U}(q_1, q_2, \ldots q_T). \tag{2.39}$$

Terms like -ln ($a_{ijik} b_{ik}(\mathbf{o}_t)$) make it easier to associate costs in going from state i_j to state i_k at time t. The weight on the path from state i to state j is $-\ln(a_{ij} b_j(\boldsymbol{o}_t))$, the negative of the logarithm of probability of going from state i to state j and selecting the observation symbol $\boldsymbol{o}_t$ in state j. The optimal sequence problem reduces to finding the path (or sequence of states) of minimum weight through which the given observation sequence occurs. The *Viterbi* algorithm is one of the best-known techniques for solving this.

2.3.5. Viterbi algorithm

Let $\mathbf{q} = (q_1, q_2, \ldots, q_T)$ be the optimal state sequence for a given observation sequence $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T)$. Then define the quantity

$$\delta_t(i) = \max_{q_1,q_2,\ldots,q_{t-1}} P[q_1 q_2 \ldots q_{t-1}, q_t = i, o_1 o_2 \ldots o_t / \lambda], \tag{2.40}$$

where $\delta_t(i)$ is the highest probability along a single path at time t, that accounts for the first t observations and ends in state i. The value of $\delta_{t+1}(j)$ can be determined through induction as:

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij}\right].b_j(\mathbf{o}_{t+1}). \tag{2.41}$$

For each t and j, the elements of the state sequence are the arguments that maximise Equation (2.42). These values are retained in the array $\psi_t(j)$ during the execution of the algorithm. The procedure for finding the best state sequence can be summarised in four steps:

- Step 1 (Initialisation)

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1), \qquad 1 \le i \le N \tag{2.42}$$

$$\psi_1(i) = 0. \tag{2.43}$$

- Step 2 (Recursion)

$$\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(\mathbf{o}_t), \qquad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N \end{matrix} \tag{2.44}$$

$$\psi_t(j) = \arg\max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] \qquad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N \end{matrix} \tag{2.45}$$

- Step 3 (Termination)

$$P^* = \max_{1 \le i \le N} \left[\delta_T(i)\right] \tag{2.46}$$

$$q_T^* = \arg\max_{1 \le i \le N} \left[\delta_T(i)\right]. \tag{2.47}$$

- Step 4 (Reconstruction)

$$q_T^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \ldots, 1. \tag{2.48}$$

The Viterbi algorithm is very similar to the forward procedure discussed previously, with the addition of Step 4. Moreover, in Step 2, the probability over previous states is maximised, unlike that of the forward procedure which involves a summation process, as is evident in Equation (2.29).

The Viterbi algorithm can also be implemented by taking the logarithms of the model parameters, thus obviating the need for numerous multiplications.

2.3.6. Problem 3 – Parameter estimation

This problem deals with training the HMM such that it encodes the observation sequence in such a way that if another observation sequence with similar characteristics to the given one is encountered later, it should be able to identify it. It is not analytically possible to solve for the model parameters in a closed form. The alternative approach is to choose the model parameters, $\lambda = (A, B, \pi)$ so that the likelihood, $P(\mathbf{O}|\lambda)$ is locally maximised. These local maxima can be found using iterative procedures or gradient techniques. Two of these techniques are discussed here, the *Baum-Welch* procedure and the *segmental K-means* algorithm.

2.3.6.1. Baum-Welch Algorithm

This approach results in parameters of the model $\lambda = (A, B, \pi)$ being adjusted so as to increase $P(\mathbf{O}|\lambda)$ until a maximum value is reached. As seen previously, calculating $P(\mathbf{O}|\lambda)$ involves summing up $P(\mathbf{O}, \mathbf{q}|\lambda)$ over all possible state sequences of $\mathbf{q}$, hence the focus is not on a particular state sequence.

The Baum-Welch (or *expectation-maximisation*, EM) method [5] maximises $P(\mathbf{O}|\lambda)$ by adjusting the parameters of λ. This optimisation criterion is called the maximum likelihood (ML) criterion, and the function $P(\mathbf{O}|\lambda)$ is called the likelihood function.

Consider the variable $\xi_t(i,j)$, which describes the probability of being in state i at time t and state j at time t+1 given the model λ and the observation sequence $\mathbf{O}$. $\xi_t(i,j)$ is defined as:

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j / \mathbf{O}, \lambda) \tag{2.49}$$

Since the joint event of the system being in state i at time t and state j at time t+1 involves both the partial observation sequences up until time t and after time t+1, $\xi_t(i,j)$ may be calculated using the forward-backward variables.

$$\begin{aligned}\xi_t(i,j) &= \frac{P(q_t = i, q_{t+1} = j/\mathbf{O},\lambda)}{P(\mathbf{O}/\lambda)} \\ &= \frac{\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{P(\mathbf{O}/\lambda)} \\ &= \frac{\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}.\end{aligned} \tag{2.50}$$

The probability of being in state i at time t given the complete observation sequence $\mathbf{O}$ and the model λ, is defined as:

$$\gamma_t(i) = P(q_t = i/\mathbf{O},\lambda). \tag{2.51}$$

$\xi_t(i,j)$ and $\gamma_t(i)$ can thus be related by summing over j,

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j). \tag{2.52}$$

Summing over the index t:

$$\sum_{t=1}^{T-1}\gamma_t(i) = \text{expected number of transitions from state } i \text{ in } \mathbf{O} \tag{2.53}$$

and

$$\sum_{t=1}^{T-1}\xi_t(i,j) = \text{expected number of transitions from state } i \text{ to state } j \text{ in } \mathbf{O}. \tag{2.54}$$

The following set of formulae to re-estimate the HMM parameters A, B and π may be obtained:

$$\bar{\pi}_j = \gamma_1(i), \tag{2.55}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}, \tag{2.56}$$

$$\bar{b}_{ij} = \frac{\sum_{t=1 \text{ such that } o_t=v_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}. \tag{2.57}$$

The descriptions of each of the above formulae are:

$$\bar{\pi}_j = \text{expected number of times in state } i \text{ at time } t=1, \tag{2.58}$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}, \tag{2.59}$$

$$\bar{b}_{ij} = \frac{\text{expected number of times in state } j \text{ and observing symbol } \mathbf{v}_k}{\text{expected number of times in state } j}. \tag{2.60}$$

The implementation of the Baum-Welch algorithm requires that a current model be defined in terms of its parameter set $\lambda = (A, B, \pi)$. Equations (2.59), (2.60) and (2.61) are applied to the current model to determine the re-estimated model $\bar{\lambda} = (\bar{A}_i, \bar{B}_i, \bar{\pi}_i)$.

It can be shown that either:

- the initial model λ is a critical point of the likelihood function, in which case $\bar{\lambda} = \lambda$, or
- $P(\mathbf{O}/\bar{\lambda}) > P(\mathbf{O}/\lambda)$ i.e. the given observation sequence O is more likely to have been produced by model $\bar{\lambda}$ rather than model λ.

It is important to bear in mind that although the result of the Baum-Welch procedure provides the ML estimate of the HMM parameters, the algorithm leads to the local maxima only.

2.3.6.2. Segmental K-means Algorithm

The segmental K-means algorithm [5] is also used to find the optimal state sequence. However, it works differently to the Viterbi algorithm. In this method, the parameters of the model $\lambda = (A, B, \pi)$ are adjusted to maximise $P(\mathbf{O}, \mathbf{q}|\lambda)$ where $\mathbf{q}$ is the optimum sequence

given by the solution to Problem 2. This criterion of optimisation is called the maximum state optimised likelihood criterion. This function $P(\mathbf{O}, \mathbf{q}^*|\lambda) = \max_{\mathbf{q}} P(\mathbf{O}, \mathbf{q}|\lambda)$ is called the state optimised likelihood function. This method of model training requires a number of observation (training) sequences. Let ω be the number of such available sequences. Each observation sequence $\mathbf{O}=\mathbf{o}_1,\mathbf{o}_2,\ldots\mathbf{o}_T$ consists of T observation symbols. Each observation symbol $\mathbf{o}_t$ is assumed to be a vector of dimension D, with $D \geq 1$. The algorithm consists of the following steps:

- Step 1

Randomly choose N observation symbols of dimension D and assign each of the ωT vectors to one of these N vectors from which its Euclidean distance is minimum. This leads to the formation of N clusters, each called a state (from 1 to N). The first vector is taken as the first vector of the first of these sequences, the second vector as the second of these sequences and so on.

- Step 2

Calculate the initial probabilities and the transition probabilities:

$$\bar{\pi}_i = \frac{\text{number of occurrences of } \{\mathbf{o}_1 \in i\}}{\text{Total number of occurrences of } \mathbf{o}_1} \qquad 1 \leq i \leq N, \tag{2.61}$$

$$\bar{a}_{ij} = \frac{\text{number of occurrences of } \{\mathbf{o}_t \in i, \text{and } \mathbf{o}_{t+1} \in j\} \text{ for all } t}{\text{number of occurrences of } \{\mathbf{o}_t \in i\} \text{ for all } t} \quad \begin{matrix} 1 \leq i \leq N \\ 1 \leq j \leq N. \end{matrix} \tag{2.62}$$

- Step 3

Calculate the mean vector and covariance vector for each state:

$$\bar{\mu}_i = \frac{1}{N_i} \sum_{\mathbf{o}_t \in i} \mathbf{o}_t \qquad 1 \leq i \leq N \tag{2.63}$$

$$\bar{\Sigma}_i = \frac{1}{N_i} \sum_{\mathbf{o}_t \in i} (\mathbf{o}_t - \bar{\mu}_i)^T (\mathbf{o}_t - \bar{\mu}_i) \qquad 1 \leq i \leq N. \tag{2.64}$$

- Step 4

Assuming a Gaussian distribution (although other density functions can also be used with no loss of generality), the symbol probability distributions for each training vector for each state is calculated:

$$\bar{b}_i(\mathbf{o}_t) = \frac{1}{(2\pi)^{D/2}\sqrt{|\bar{\Sigma}_i|}} \exp\left[-\frac{1}{2}(\mathbf{o}_t - \bar{\mu}_i)\bar{\Sigma}_i^{-1}(\mathbf{o}_t - \bar{\mu}_i)^T\right] \quad 1 \le i \le N. \quad (2.65)$$

- Step 5

  Find the optimal state sequence **q***(as given by the solution to Problem 2) for each training sequence using $\bar{\lambda} = (\bar{A}_i, \bar{B}_i, \bar{\pi}_i)$calculated in Steps 2 – 4. A vector is reassigned a state if its original assignment is different from the corresponding estimated optimum state, i.e. for all training sequences, assign $\mathbf{o}_t$ (of say the k^{th} training sequence) to state *i* if q^*_t is state *i*.

- Step 6

  If any vector is assigned a new state as a result of Step 5, use the new assignment and repeat Steps 2 – 6; otherwise, stop.

Once the optimal state sequence has been determined, the HMMs must be updated. Once such technique, incorporating embedded model re-estimation algorithms, are discussed in the next section.

2.3.7. Embedded model re-estimation

The concept of embedded model re-estimation is discussed in [9, 49, 52] and is based on its implementation in the Hidden Markov Model Toolkit (HTK) [9]. The concept as it is applied in HTK, updates all of the HMMs in a system using all of the training data in a single iteration. A complete description of the HTK toolkit is found in Appendix A.

A complete set of HMM definitions are loaded on initialisation. Every training file must have an associated label file that gives a transcription for that file. Only the sequence of labels is used and boundary location information is ignored. Thus, these transcriptions can be generated automatically from the known orthography of what was said and a pronunciation dictionary. The embedded model re-estimation algorithm processes each training file in turn. The associated transcription is then used to construct a composite HMM which spans the whole utterance. This composite HMM is made by concatenating instances of the phone HMMs corresponding to each label in the transcription. The forward-backward algorithm is then applied and the sums needed to form the weighted averages accumulated in the standard

way. When all of the training files have been processed, the new parameter estimates are formed from the weighted sums and the updated HMM set is output.

Once the HMMs have been updated, further statistical techniques are utilised to build the speech recogniser. Statistical language models that include the syntactic constraints of the language are introduced next.

2.3.8. Statistical language models

Statistical language models have become a key point in the speech recognition systems. The language model is the recognition system component that incorporates the syntactic constraints of the language. Most statistical language models are based on the empirical paradigm that a good estimation of the probability of a linguistic event can be obtained by observing this event on a large enough text corpus. The most commonly used models are *n-grams*, where the probability of a basic unit (phoneme, syllable, word, etc.) is estimated from the conditional probabilities of each basic unit given the *n*-1 preceding units. While these models are both robust and efficient, they are limited to modeling only the local linguistic structure and are difficult to estimate for all but *n*=2 or *n*=3 [5]. Hence *bi-gram* and *tri-gram* language models are widely used in speech recognition systems.

In this study, the backed-off bi-gram (see description below) is calculated using the formula [9]:

$$p(i,j) = \begin{cases} \dfrac{N(i,j)-d}{N(i)} & \text{if } N(i,j) > t \\ b(i)\,p(j) & \text{otherwise.} \end{cases} \qquad (2.66)$$

N(i,j) is the number of times phone *j* follows phone *i*, *N(i)* is the number of times that phone *i* appears. A process called *discounting* [9] is used whereby a small probability (usually *d*=0.5) of the available probability mass is deducted from the higher bi-gram counts and distributed amongst the infrequent bi-grams. When the bi-gram count falls below the threshold *t*, the bi-gram is backed-off to the uni-gram probability scaled by a backed-off weight. This ensures that all bi-gram probabilities for a given history total one.

An important issue in the development of speech recognition systems is how to create language models for spontaneous speech. While this is not relevant to the present investigation, it is of interest to note that for spontaneous speech, it is necessary to deal with extraneous words, out-of-vocabulary words, hesitations, repetitions, ungrammatical sentences and even partial words. This kind of variation can degrade the recognition performance.

Statistical language models, that are used to improve the recogniser's speech recognition abilities, were described in this section. Speaker adaptation techniques have also been extensively used to improve recognition rates. Speaker adaptation techniques have been traditionally used to move from the speaker independent to the speaker dependent case. However, they have been used extensively as one of the key components of multilingual speech recognition as well. The theory behind these techniques is discussed next.

2.4. Speaker adaptation theory and techniques

The traditional idea behind speaker adaptation is to use a small amount of adaptation data to change the recognition system such that it models as much of the speaker-specific information as possible. Many approaches have been developed which try to produce this effect. Speaker adaptation techniques for HMM-based recognition systems fall into two basic categories. The first of these employs techniques that transform the input speech of the new speaker to a vector space that is common with the training speech. These are known as *spectral mapping* techniques. The second category consists of methods that transform the model parameters to better match the characteristics of the adaptation data. These techniques are known as *model mapping* approaches.

The spectral mapping approach is based on the belief that a recognition system can be improved by matching the new speaker's features vectors to the vectors of the training data [20]. The mapping is designed so that the difference between the reference vector set and the mapped vector set is minimised. These differences are due to the spectral differences of the speakers' speech production systems (such as vocal tract length and shape).

Initial attempts at spectral mapping adaptation were used in the spectral template matching systems [21]. These consider the template to be from the reference speaker and automatically generate a transformation to minimise the difference between the new speaker and the reference speaker. Another method that is similar to speaker normalisation uses a transform to map each speaker in the speaker-independent training set onto a reference speaker [22]. Thus, the models generated act as speaker-dependent models.

Spectral mapping techniques aim to improve the match between the reference speaker and new speaker. However, this goal does not explicitly try to increase the accuracy of the models for the new speaker. This means that it does not take full advantage of the adaptation data. This is an area addressed by the model-mapping approach. Rather than trying to map all speakers to one space, the model-mapping approach adjusts the model parameters to best represent the new speaker.

Two issues that must be addressed when discussing model-mapping approaches are the training modes (supervised vs. unsupervised) and the adaptation mode (incremental verses batch). In a supervised training mode, the recognition system is given the correct transcription and has only to align the user's speech to that transcription. In unsupervised adaptation the recogniser feeds itself, perhaps including recognition errors. The supervised mode is preferred when available.

The adaptation mode describes when the adaptation takes place and what models are employed to produce the hypotheses used for adaptation. In incremental mode, the models are adapted quite often and the adapted models are used to produce the hypotheses for the next adaptation. This is the typical method seen in real-time systems that use adaptation. Batch mode is similar to a training run where hypotheses for the entire adaptation set are stored and then used to iteratively update the adapted models.

2.4.1. Maximum Likelihood Linear Regression (MLLR)

Maximum likelihood linear regression (MLLR) is a technique that computes a set of transformations that aims to reduce the mismatch between an initial model set and the adaptation data by estimating a set of linear transformations for the mean and variance parameters of a Gaussian mixture HMM system. These transformations shift the component means and alter the variances in the initial system so that each state in the HMM system is more likely to generate the adaptation data.

The fundamental idea behind MLLR is to tie or cluster some Gaussian mixtures together in order to reduce the number of parameters to be updated and force the mixtures to share the same adaptation matrix.

The theory that is discussed in this section is summarised from [58].

Let **W** be the transformation matrix used to give a new estimate of the adapted mean, which in turn is given by:

$$\hat{\boldsymbol{\mu}} = \mathbf{W}\boldsymbol{\xi}, \tag{2.67}$$

where the adaptation data is of dimension n, **W** is the n x (n + 1) transformation matrix, and $\boldsymbol{\xi}$ is the extended mean vector,

$$\boldsymbol{\xi} = [w\ \mu_1\ \mu_2 \ldots \mu_n]^T, \tag{2.68}$$

with w representing the offset indicator.

For Gaussian probability models, this gives an adapted mixture density of

$$\bar{b}_s(\mathbf{o}_t) = \frac{1}{(2\pi)^{D/2}\sqrt{|\bar{\Sigma}_s|}} \exp\left[-\frac{1}{2}(\mathbf{o}_t - \mathbf{W}\boldsymbol{\xi})\bar{\Sigma}_s^{-1}(\mathbf{o}_t - \mathbf{W}\boldsymbol{\xi})^T\right] \tag{2.69}$$

The transformation matrix **W** is obtained by solving a maximisation problem using the Expectation-Maximisation (EM) technique. In a similar way, the EM technique is again used to compute the variance transformation matrix. Using EM results in the maximisation of a

standard auxiliary function. For speech recognition systems, the auxiliary function that displays good convergence properties and is thus often used is given by:

$$Q\left(\lambda,\hat{\lambda}\right) = \sum_{\mathbf{q}} P(\mathbf{O},\mathbf{q}/\lambda)\log\left\{P\left(\mathbf{O},\mathbf{q}/\hat{\lambda}\right)\right\} \tag{2.70}$$

where $\hat{\lambda}$ is the transformed model and $\mathbf{q}$ contains all possible state sequences leading to the recognition of $\mathbf{O}$.

For HMMs, the probabilities are related to both the transition probabilities and the state emission probabilities. This means that the auxiliary function can be expanded as:

$$Q\left(\lambda,\hat{\lambda}\right) \propto \sum_{\mathbf{q}} P(\mathbf{O},\mathbf{q}/\lambda)\left[\sum_{t=1}^{T}\log\left(\hat{transition\ prob.}\right)+\sum_{t=1}^{T}\log\left(\hat{b}_{q_t}(\mathbf{o}_t)\right)\right] \tag{2.71}$$

The terms in the above equation due to transition probabilities can be ignored since the primary goal is to re-estimate the transformation matrix. This leads to an auxiliary function of the form:

$$Q\left(\lambda,\hat{\lambda}\right) \propto \text{constant} + \sum_{\mathbf{q}}\sum_{t=1}^{T} P(\mathbf{O},\mathbf{q}/\lambda)\log\left(\hat{b}_{q_t}(\mathbf{o}_t)\right) \tag{2.72}$$

The posterior probability of occupying state *s* at time *t* given that the observation sequence **O** is generated is defined as:

$$\gamma_s(t) = \frac{1}{P(\mathbf{O}/\lambda)}\sum_{\mathbf{q}} P(\mathbf{O},\mathbf{q}_t = s/\lambda) \tag{2.73}$$

Let *S* be the set of all states in the system. The total probability can be obtained by summing the marginal probabilities across all states, making the auxiliary function:

$$Q\left(\lambda,\hat{\lambda}\right) \propto \text{constant} + P(\mathbf{O}/\lambda)\sum_{j=1}^{S}\sum_{t=1}^{T}\gamma_j(t)\log\left(\hat{b}_j(\mathbf{o}_t)\right). \tag{2.74}$$

To maximise $Q\left(\lambda,\hat{\lambda}\right)$, its derivative with respect to $\mathbf{W}$ is computed and equated to zero, i.e.

$$\frac{dQ(\lambda,\hat{\lambda})}{d\mathbf{W}} = P(\mathbf{O}/\lambda)\frac{d}{d\mathbf{W}}\sum_{j=1}^{S}\sum_{t=1}^{T}\gamma_j(t)\log\left(\hat{b}_j(\mathbf{o}_t)\right). \tag{2.75}$$

Expanding $\hat{b}_j(\mathbf{o}_t)$ in a Gaussian reduces the above differential to:

$$\frac{dQ(\lambda,\hat{\lambda})}{d\mathbf{W}} = -\frac{1}{2}P(\mathbf{O}/\lambda)\frac{d}{d\mathbf{W}}\sum_{j=1}^{S}\sum_{t=1}^{T}\gamma_j(t)\left[n\log(2\pi)+h(\mathbf{o}_t,j)+\log(\Sigma_j)\right], \tag{2.76}$$

where

$$h(\mathbf{o}_t,j) = (\mathbf{o}_t-\mathbf{W}_j\xi_j)\Sigma_j^{-1}(\mathbf{o}_t-\mathbf{W}_j\xi_j)^t. \tag{2.77}$$

Since $h(\mathbf{o}_t,s)$ is the only term in the summation that is dependent on W, the differential reduces to:

$$\frac{dQ(\lambda,\hat{\lambda})}{d\mathbf{W}} = -\frac{1}{2}P(\mathbf{O}/\lambda)\sum_{t=1}^{T}\gamma_s(t)\frac{d}{d\mathbf{W}}h(\mathbf{o}_t,s), \tag{2.78}$$

or

$$\frac{dQ(\lambda,\hat{\lambda})}{d\mathbf{W}} = -\frac{1}{2}P(\mathbf{O}/\lambda)\sum_{t=1}^{T}\gamma_s(t)\Sigma_s^{-1}(\mathbf{o}_t-\mathbf{W}_s\xi_s)\xi_s{}^t. \tag{2.79}$$

This function is optimised by setting it to zero, and rearranging terms gives:

$$\sum_{t=1}^{T}\gamma_s(t)\Sigma_s^{-1}\mathbf{o}_t\xi_s{}^t = \sum_{t=1}^{T}\gamma_s(t)\Sigma_s^{-1}\mathbf{W}_s\xi_s\xi_s{}^t. \tag{2.80}$$

This is the general form for computing $\mathbf{W}_s$. The closed-form solution can be found when all covariance matrices are diagonal. If $\mathbf{W}_s$ is shared by R states, $\{s_1,s_2, \ldots, s_R\}$, then the general form becomes:

$$\sum_{t=1}^{T}\sum_{r=1}^{R}\gamma_{s_r}(t)\Sigma_{s_r}^{-1}\mathbf{o}_t\xi_{s_r}{}^t = \sum_{t=1}^{T}\sum_{t=1}^{T}\gamma_{s_r}(t)\Sigma_{s_r}^{-1}\mathbf{W}_s\xi_{s_r}\xi_{s_r}{}^t. \tag{2.81}$$

This can be rewritten as:

$$\sum_{t=1}^{T}\sum_{r=1}^{R}\gamma_{s_r}(t)\Sigma_{s_r}^{-1}\mathbf{o}_t\xi_{s_r}{}^t = \sum_{r=1}^{R}\mathbf{V}^{(r)}\mathbf{W}_s\mathbf{D}^{(r)}, \tag{2.82}$$

where

$$\mathbf{V}^{(r)} = \sum_{t=1}^{T} \gamma_{s_r}(t) \Sigma_{s_r}^{-1}, \tag{2.83}$$

and $\mathbf{V}^{(r)}$ is the state distribution inverse covariance matrix scaled by the state occupancy probability, and where

$$\mathbf{D}^{(r)} = \xi_{s_r} \xi_{s_r}{}^{t}, \tag{2.84}$$

and $\mathbf{D}^{(r)}$ is a singular matrix calculating the outer product of the extended mean vector.

Let the right hand side of Equation (2.84) be an n x (n + 1) matrix, **Z**, and let the elements of **Z**, $\mathbf{V}^{(r)}$, $\mathbf{W}_s$, $\mathbf{D}^{(r)}$ be y_{ij}, $v_{ij}{}^{(r)}$, w_{ij} and $d_{ij}{}^{(r)}$ respectively. This reduces the equation to:

$$z_{ij} = \sum_{p=1}^{n} \sum_{q=1}^{n+1} w_{pq} \sum_{r=1}^{R} v_{ip}^{(r)} d_{qj}^{(r)}. \tag{2.85}$$

$\mathbf{D}^{(r)}$ is symmetric since all covariances are diagonal, then:

$$\sum_{r=1}^{R} v_{ip}^{(r)} d_{qj}^{(r)} = \begin{bmatrix} \sum_{r=1}^{R} v_{ii}^{(r)} d_{qj}^{(r)} & \text{when } i = p \\ 0, & \text{when } i \neq p \end{bmatrix} \tag{2.86}$$

and

$$z_{ij} = \sum_{q=1}^{n+1} w_{iq} \sum_{r=1}^{R} v_{ii}^{(r)} d_{jq}^{(r)}. \tag{2.87}$$

Setting

$$g_{jq}^{(i)} = \sum_{r=1}^{R} v_{ii}^{(r)} d_{jq}^{(r)} \tag{2.88}$$

gives

$$z_{ij} = \sum_{q=1}^{n+1} w_{iq} g_{jq}^{(i)}, \tag{2.89}$$

where $g_{jq}^{(i)}$ are the elements of an (n+1) x (n+1) matrix, $\mathbf{G}^{(i)}$. Since $\mathbf{D}^{(r)}$ is singular, $\mathbf{G}^{(i)}$ is also singular. z_{ij} and $g_{jq}^{(i)}$ can be computed from the observation vectors and the model parameters. This gives a set of linear re-estimation equations

$$\mathbf{w}_i^t = \left(G^{(i)}\right)^{-1} z_i^t \tag{2.90}$$

where $\mathbf{w}_i$ and $\mathbf{z}_i$ are the i^{th} rows of $\mathbf{W}_s$ and **Z** respectively.

For the covariance transform estimation, first define the adapted variance as:

$$\hat{\boldsymbol{\Sigma}}_s = \mathbf{B}_s^t \hat{\mathbf{H}}_s \mathbf{B}_s, \tag{2.91}$$

where $\hat{\mathbf{H}}_s$ is the transform to be estimated and $\mathbf{B}_s$ is the inverse of the Cholesky factor of $\boldsymbol{\Sigma}_s^{-1}$. Hence,

$$\boldsymbol{\Sigma}_s^{-1} = \mathbf{C}_s \mathbf{C}_s^t \tag{2.92}$$

and

$$\mathbf{B}_s = \mathbf{C}_s^{-1}. \tag{2.93}$$

Cholesky decomposition ensures that the resulting matrix is non-singular. The same auxiliary function as previously defined in Equation (2.74) is used, i.e.

$$Q(\lambda, \hat{\lambda}) \propto \text{constant} + P(\mathbf{O}/\lambda) \sum_{j=1}^{S} \sum_{t=1}^{T} \gamma_j(t) \log\left(\hat{b}_j(\mathbf{o}_t)\right). \tag{2.94}$$

Expanding $\log\left(\hat{b}_j(\mathbf{o}_t)\right)$ using Equations (2.71) and (2.91) yields:

$$\log\left(\hat{b}_j(\mathbf{o}_t)\right) = -\frac{1}{2}\left[n\log(2\pi) + \log\left(\boldsymbol{\Sigma}_j\right)\right] - \frac{1}{2}\log\left|\hat{\mathbf{H}}_j\right| - \frac{1}{2}\left[\left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_j\right)^t \mathbf{B}_j^{-1} \hat{\mathbf{H}}_j^{-1} \left(\mathbf{B}_j^t\right)^{-1} \left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_j\right)\right] \tag{2.95}$$

Since $\mathbf{B}_j = \mathbf{C}_j^{-1}$,

$$\log\left(\hat{b}_j(\mathbf{o}_t)\right) = -\frac{1}{2}\left[n\log(2\pi) + \log\left(\boldsymbol{\Sigma}_j\right)\right] - \frac{1}{2}\log\left|\hat{\mathbf{H}}_j\right| - \frac{1}{2}\left[\left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_j\right)^t \mathbf{C}_j \hat{\mathbf{H}}_j^{-1} \mathbf{C}_j^t \left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_j\right)\right] \tag{2.96}$$

or

$$\log\left(\hat{b}_j(\mathbf{o}_t)\right) = -\frac{1}{2}\left[n\log(2\pi) + \log\left(\boldsymbol{\Sigma}_j\right)\right] - \frac{1}{2}\log\left|\hat{\mathbf{H}}_j\right| - \frac{1}{2}\left[\left(\mathbf{C}_j^t\mathbf{o}_t - \mathbf{C}_j^t\hat{\boldsymbol{\mu}}_j\right)^t \hat{\mathbf{H}}_j^{-1} \left(\mathbf{C}_j^t\mathbf{o}_t - \mathbf{C}_j^t\hat{\boldsymbol{\mu}}_j\right)\right] \tag{2.97}$$

To maximise $Q(\lambda, \hat{\lambda})$, its derivative with respect to $\hat{\mathbf{H}}_s$ is computed and equated to zero. Grouping like terms gives:

$$\hat{\mathbf{H}}_s = \frac{\mathbf{C}_s^t \sum_{t=1}^{T} \gamma_s(t)\left[\left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_s\right)\left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_s\right)^t\right]\mathbf{C}_s}{\sum_{t=1}^{T} \gamma_s(t)}. \tag{2.98}$$

If $\hat{\mathbf{H}}_s$ is shared by R states, $\{s_1, s_2, \ldots, s_R\}$, then

$$\hat{\mathbf{H}}_s = \frac{\sum_{r=1}^{R} \mathbf{C}_{s_r}^t \sum_{t=1}^{T} \gamma_{s_r}(t) \left[(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_s)(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_{s_r})^t \right] \mathbf{C}_{s_r}}{\sum_{r=1}^{R} \sum_{t=1}^{T} \gamma_{s_r}(t)}. \quad (2.99)$$

The transformation of the covariance using the estimate for $\hat{\mathbf{H}}_s$ results in a full covariance matrix, but the off-diagonal terms in $\hat{\mathbf{H}}_s$ can be set to zero and an increase in likelihood is still guaranteed.

2.4.2. Maximum A Posteriori (MAP) adaptation

The maximum a posteriori (MAP) approach is another technique that performs model adaptation. This adaptation process is also known as Bayesian adaptation. MAP adaptation involves using prior knowledge about the model parameter distribution. This has the advantage that the prior knowledge gives an idea of what the parameters of the model are most likely to be, thus allowing optimum use of the limited adaptation data during the training process. This type of prior is known as an informative *prior*. If the prior distribution does not indicate what the model parameters are most likely to be, it is termed a *non-informative* prior. When non-informative priors are used, the MAP estimate obtained will be identical to that obtained using a maximum likelihood approach.

The theory that is discussed in this section is summarised from [59].

In the discussion that follows, let the sample $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be a given set of n observations where each observation is drawn from the probabilistic function of a Markov chain.

Assume that θ is the parameter vector to be estimated from the sample vector $\mathbf{x}$ with a probability density function $f(.|\theta)$. Assume that θ is a random vector that takes it values in the space Θ and if g is the prior density function of θ then the map estimate θ_{MAP} is defined as the mode of the posterior density function of θ [24,25],

$$\theta_{MAP} = \arg\max_{\theta} f(\mathbf{x}/\theta)g(\theta) \quad (2.100)$$

If θ is assumed to be fixed but unknown, then there is no knowledge about θ, hence the assumption of a non-informative prior is used i.e. $g(\theta)$=constant. The above equation then reduces to the maximum likelihood formulation.

The key problems for the MAP formulation are:

- the choice of the prior distribution family, and
- the evaluation of the maximum a posteriori.

The appropriate choice of prior distribution can simplify the MAP estimation. The MAP estimation is easier if the family of density functions possesses a sufficient statistic of fixed dimension, however this is only true for exponential families.

Consider an N-mixture Gaussian density CDHMM with parameters $\{\mu, \Sigma\}$ for every observation density, where

$$\mu = (\mu_1, \mu_2, \ldots, \mu_N), \quad (2.100)$$

$$\boldsymbol{\Sigma} = (\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \ldots, \boldsymbol{\sigma}_N). \quad (2.101)$$

Here the parameters $\{u_i, \boldsymbol{\sigma}_i\}$ are the mean vector and the covariance matrix of the i^{th} Gaussian mixture component of an observation density. The idea is to find a transformation from the original HMM to an HMM that closer resembles the observation data. The transformed HMM's observation parameters $\{m, \gamma\}$ are as follows:

$$m = A * \mu, \quad (2.102)$$

$$\gamma = B * \Sigma, \quad (2.103)$$

where * is a mathematical operator, A and $\mathbf{B}$ are transformations.
From [24, 25], the transformation of the k^{th} Gaussian density of the i^{th} state of an HMM is therefore as follows:

$$m_k = \frac{\tau_k \mu_k + \sum_{t=1}^{T} \theta_t(k) o_t}{\tau_k \mu_k + \sum_{t=1}^{T} \theta_t(k)}, \quad (2.104)$$

$$\gamma_k = \frac{\beta_k + \sum_{t=1}^{T} \theta_t(k)(o_t - \mu_k)(o_t - \mu_k)^T}{(\alpha_k - p)\sum_{t=1}^{T} \theta_t(k)} + \frac{\tau_k \cdot (m_k - \mu_k)(m_k - \mu_k)^T}{(\alpha_k - p) + \sum_{t=1}^{T} \theta_t(k)}, \quad (2.105)$$

where,

$$\theta_t(k) = \delta(s_t - i)\frac{\omega_k \aleph(o_t / \mu_k, \sigma_k)}{\sum_{n=1}^{N} \omega_k \aleph(o_t / \mu_n, \sigma_n)}. \quad (2.106)$$

$\theta_t(k)$ is the probability of the model generating $\mathbf{o}_t$ while being in the i^{th} state with mixture component label k at time t. δ denotes the Kronecker delta function, which returns a value of 1 when the optimal state sequence $S=\{s_1,s_2,...,s_T\}$ (determined by Viterbi segmentation) is arrived at in the i^{th} state at time t; it returns a value of 0 otherwise.

In this chapter, the background theory relating to general speech recognition tasks was discussed. An overview of speech sounds and how they are produced was presented. The pre-processing and feature extraction techniques that are used were covered. HMMs were discussed in detail. The theory behind speaker adaptation was then presented, and the MLLR and MAP adaptation techniques were covered. The next chapter discusses selected distance measures that were used together with these adaptation techniques in this cross-language phoneme mapping investigation.

Chapter 3

ACOUSTIC DISTANCE MEASURES

A variety of distance formulations exist to compute the distances between Gaussian distributions obtained for each phoneme model. The distance measures chosen for this investigation have shown numerous applications in other pattern recognition studies.

This chapter centres around the theory behind selected distance measures. In the sections that follow, let $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ represent the feature mean vector and covariance matrix respectively for a Gaussian distribution *i*.

3.1. Kullback-Leibler distance

A popular distance metric used previously in calculating the distance between two models is the Kullback-Leibler measure [1, 4, 5, 19]. It has been used extensively in pattern recognition applications to judge how close two probability distributions are. The Kullback-Leibler distance is given by:

$$D_{KL} = \frac{1}{2}(\mu_2 - \mu_1)^T \left[\Sigma_1^{-1} + \Sigma_2^{-1}\right](\mu_2 - \mu_1) + \frac{1}{2} tr\left(\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 - 2I\right). \quad (3.1)$$

3.2. Bhattacharyya distance

The Bhattacharyya distance metric [2, 3] has been extensively used to obtain the distance between phoneme models of different languages. Mak and Barnard [2] have shown that the Bhattacharyya distance can be effectively used in phone clustering applications. This distance measure is given by:

$$D_{Bha} = \frac{1}{8}(\mu_2 - \mu_1)^T \left[\frac{\Sigma_1 + \Sigma_2}{2}\right]^{-1} (\mu_2 - \mu_1) + \frac{1}{2}\log \frac{\left|\frac{\Sigma_1 + \Sigma_2}{2}\right|}{\sqrt{|\Sigma_1||\Sigma_2|}}. \tag{3.2}$$

The first term gives the class separability as a result of the class means, while the second term gives the class separability between the class covariance matrices.

3.3. Mahalanobis measure

The Mahalanobis distance metric has also been used as a distance classifier. It has the advantage that by utilising the information available in the covariance matrices, it takes the variability between the models to be compared into account. The Mahalanobis distance [1] is given by the equation:

$$D_{Mah} = \frac{1}{n}(\mu_2 - \mu_1)^T (\Sigma_1 \Sigma_2)^{-1} (\mu_2 - \mu_1). \tag{3.3}$$

3.4. Euclidean measure

The one-dimensional Euclidean measure has previously been used to calculate inter-class distances [1, 4, 5]. This geometric measure is given by:

$$D_{Euc} = \sqrt{(\mu_2 - \mu_1)^T (\mu_2 - \mu_1)}. \tag{3.4}$$

3.5. L2 metric

Another popular measure is the L2 distance [1]. In the general case, this distance measure is represented as:

$$D_{L2} = \sqrt{\int_{\Re^n} \left[N_2(\overline{\mu}_2, \Sigma_2) - N_1(\overline{\mu}_1, \Sigma_1) \right]^2 d\overline{X}} \,, \tag{3.5}$$

where $N_1(\overline{\mu}_1, \Sigma_1)$ and $N_2(\overline{\mu}_2, \Sigma_2)$ represent two Gaussian distributions.

A closed form of the L2 distance measure exists if Gaussian distributions are assumed. The L2 distance measure then reduces to:

$$(D_{L2})^2 = \prod_{k=1}^{n} \frac{1}{2\sigma_{1k}\sqrt{\pi}} + \prod_{k=1}^{n} \frac{1}{2\sigma_{2k}\sqrt{\pi}} + 2\prod_{k=1}^{n} \frac{1}{\sqrt{2\pi}\sqrt{\sigma_{1k}^2 + \sigma_{2k}^2}} \exp\left[\frac{1}{2} \frac{\left(\mu_{1k} \frac{\sigma_{2k}}{\sigma_{1k}} + \mu_{2k} \frac{\sigma_{1k}}{\sigma_{2k}} \right)^2}{\sigma_{1k}^2 + \sigma_{2k}^2} - \left(\frac{\mu_{1k}^2}{\sigma_{1k}^2} + \frac{\mu_{2k}^2}{\sigma_{2k}^2} \right) \right]. \tag{3.6}$$

3.6. Jeffreys-Matusita distance

The final distance measure that is used during this investigation is the Jeffreys-Matusita distance measure [10, 28], which is closely related to the Bhattacharyya distance. Huber and Mayer [27] have previously used the Jeffreys-Matusita distance metric in an image processing application to identify the most relevant features among a large number of texture energy features derived from synthetic aperture radar images. Swain and King [28] successfully applied the Jeffreys-Matusita distance measure as a feature selection criterion for remote sensing applications where determining interclass separability is critical.

If $p_1(x)$ and $p_2(x)$ are the conditional probability density functions, then the general form of the

Jeffreys-Matusita distance can be written as:

$$D_{JM} = \sqrt{\int_x \left[\sqrt{p_1(x)} - \sqrt{p_2(x)}\right]^2 dx}\,. \tag{3.7}$$

It reduces to the following expression if a Gaussian distribution is used:

$$D_{JM} = \sqrt{2\left(1 - e^{-\alpha}\right)} \tag{3.8}$$

where α is given by the value of the Bhattacharyya distance in Equation (3.2).

This chapter has focused on the acoustic distance measures that were used during the course of the investigation. The following chapter deals with the experimental protocol that was used for the experiments.

Chapter 4

EXPERIMENTAL PROTOCOL

This chapter describes the experimental protocol that was used during the investigation. The first section gives an overview of the methodology. The tools that were used are also described in more detail. The general recogniser training methodology is then covered. The configuration parameters that can be configured in the Hidden Markov Toolkit are then described, together with their values used. The performance criteria that were used during the course of the investigation to assess the recognisers' phoneme recognition abilities are then given. Finally, the test and training database particulars are discussed.

4.1. Overview of Experimental protocol

The Hidden Markov Toolkit (HTK) version 3 [9] was the primary tool utilised to conduct all the experiments. HTK contains built-in functions that implement MLLR and MAP adaptation as well as Baum-Welch re-estimation. These functions were used during this study. More information on HTK can be found in Appendix A. HTK provides a comprehensive set of functions that are needed in order to conduct experiments in speech recognition. Perl scripting and MATLAB programming were used to supplement the functionalities of HTK.

The configuration parameters that determine which functions are used and field values for attributes are stored in configuration files. The directory locations for the speech data are also

configured in these files. A list of the training and test data are found in separate configuration files. The actual speech data that is analysed by HTK is in WAV format. The WAV file format is a subset of Microsoft's RIFF specification for the storage of multimedia files. The labelled phonetic data for each speech file is found in text files containing the start and end time of each phonetic sound, and the associated label.

The speech recognisers were separately trained on the training data for a pre-determined number of iterations not exceeding a maximum value specified in the configurations files. The training and test phases are initiated by means of scripts. At the completion of the training phase, the testing phase was started. HTK's statistical tools were then used to determine the performance of the recogniser.

The HMM parameters (the mean and variance) relating to each phoneme model are stored as text files in a directory designated in the configuration files. Perl scripting was used to process the HMM parameter files into a format that could be used by MATLAB. The distance calculations between the phonemes were then carried out in MATLAB.

Once the cross-language phoneme mapping matrix was calculated and the nearest phoneme was determined, it was necessary to change the phoneme label files. HTK does have a tool capable of replacing phoneme labels. However, it was found that this could just as easily be carried out using Perl scripting.

Once the speech phoneme data had been relabeled, it could then be used to:

- Re-test the trained recogniser on the re-labelled phoneme data;
- Adapt the recogniser's models using MLLR or MAP adaptation;
- Re-train the recogniser on pooled speech data (including the re-labelled phoneme data). The recogniser's phoneme recognition abilities were then re-tested as before.

The next section describes the general recogniser training methodology in more detail.

4.2. General recogniser training methodology

The procedure below refers to the general training methodology for a recogniser.

4.2.1. Building the language model

Using the training transcriptions, the bi-gram phone probabilities are estimated, i.e. $P(j|i)$, the probability of phone *j* being followed by phone i.

Once the task grammar is defined, the phone network is built from the bi-gram phone probabilities. This identifies phones and their contextual relationship and is stored in Standard Lattice Format (SLF) in HTK using Extended Backus Naur form grammar notation.

4.2.2. Model initialisation

HMM topology prototype and feature files are used to clone the HMM models.

Models are initialised using global mean and covariances (flat start). The phonetically transcribed speech data is then used to bootstrap monophone models. The segmental K-means algorithm is then used to generate the initial HMM model estimates.

4.2.3. Model retraining

Iterative Baum-Welch re-estimation of the parameters of a single HMM is performed using the training data. This process is repeated after each increase in the number of mixture components.

4.2.4. Viterbi realignment

Since there are multiple pronunciations per phoneme, Viterbi decoding is used to select the best-match pronunciation hypothesis. This constrains the search to the most likely candidate

phonemes. A Viterbi score for each candidate is weighted by the bi-gram probability. A beam search is performed to prune low probability paths. Paths that have a score less than {maximum score – threshold} are deleted.

This section has dealt with the general methodology used to train the recogniser. The next section covers the configuration parameters within the HTK toolkit.

4.3. Configuration parameters within speech toolkit

For the majority of the experiments, single-state HMMs are used. Note that in HTK, single-state HMMs are modelled as three-state HMMs. This is done to cater for the non-emitting ENTER and EXIT states defined in HTK.

One set of experiments (to determine the impact of the number of HMM states on cross-language phoneme recognition performance) does utilise three-state, left-to-right Continuous Density Hidden Markov Models that do not allow skip transitions.

Gaussian probability density functions were utilised. Wherever necessary, diagonal covariance matrices were employed. Initially, single-mixture Gaussian pdfs were used. This was increased in steps of one to four mixture components. Whenever mixture components are found, only the dominant mixture component per state is used in the distance calculation.

A Hamming window was used during the pre-processing, with the window size for all experiments set to 25.6 ms, and the frame size to 10 ms. A pre-emphasis filter coefficient of 0.97 was used.

The HTK configuration file was set up to calculate 12 Mel-Frequency Cepstral Coefficients (MFCCs) and a log energy measure. The latter was included since it has been previously shown that the energy of an utterance contains important information about the phonetic identity of the utterance [5]. The number of filters in the mel-scale filter bank was kept constant at 26. Energy normalisation was also included. The impact on the recogniser performance of including delta coefficients and acceleration coefficients was also investigated.

Cepstral mean normalisation [9] was performed to compensate for audio effects. This is especially relevant in this set of experiments where two independent speech databases are compared.

Cepstral *liftering* or rescaling was performed to compensate for the high variance in the magnitude of the cepstral coefficients.

Now that the configuration parameters within the toolkit have been discussed, the performance criteria used to evaluate the recognisers' performance are given next.

4.4. Performance Criteria

Different acoustic measures are used to compute the acoustic similarity between the TIMIT phoneme models and the SUN Speech phoneme models. The six different distance measures described in Chapter 2 are used. Since the Gaussian models are multi-mixture (up to four mixtures per state were used in the experiments) and single-state, the distance between two phones was calculated using the most dominant mixture component in the state. The investigation also looks into the effect of using multi-state (three-state) HMMs. In this set of experiments, only the middle state is considered for the acoustic distance calculation.

The automated approach is then compared to a manual phoneme-mapping procedure carried out by a phonetic expert [3]. Automated mappings from English to Afrikaans phonemes, and vice versa are investigated. The quality of the mapping is determined from the cross-database (i.e. cross-language) recognition performance.

In the experiments that were conducted, the following performance criteria are used:

$$\%\,\text{Correct labels} = \frac{\text{number of correct labels}}{\text{total number of labels}} \text{ X } 100\% \qquad (4.1)$$

and,

$$\%\,\text{Accuracy} = \frac{\text{number of correct labels} - \text{insertions}}{\text{total number of labels}} \text{ X } 100\%. \qquad (4.2)$$

The next section describes the details of the training and test databases used during the investigation.

4.5. Training and Test Database Particulars

The experiments are carried out using the TIMIT English database [11] and the SUN Speech English-Afrikaans corpus [12]. Only the SI (phonetically-diverse) and SX (phonetically-compact) TIMIT sentence sets were used. The TIMIT database contains about 80% more speech data than the English part of the SUN Speech database. There are 39 different phonemes listed in the TIMIT database (including the silence model) and a total of 59 phonemes used in the labelling of the SUN Speech database. Additional information on the TIMIT and SUN Speech corpora can be found in Appendix B.

For the purposes of these experiments the [cl] silence model in TIMIT and the [sil] model in the SUN Speech database are mapped to each other. However, as an aside, these non-voice models were also included as part of the mapping experiments and their nearest neighbour phoneme models can be found in Appendices C and D.

Moreover, 6 phoneme classes are found only in the English segment of the SUN Speech corpus, not in Afrikaans. Since only the Afrikaans data was used, the phoneme-mapping experiments involve 38 TIMIT "base" or "reference" phonemes and 52 SUN Speech phonemes.

For the purposes of the experiment, only the Afrikaans part of the SUN Speech database was utilised. This was done to mimic practical instances where a small amount of data is available for the new language (Afrikaans) and where a fully trained recogniser already exists for a base language (English).

The SUN Speech database consists of two phonetically rich sentence sets (693 sentences in total) spoken by male and female speakers. The database consists of speakers who spoke both sets and either one of the two sets. Table 4.1 summarises this breakdown.

Table 4.1: Details of the SUN Speech Afrikaans database

Sentence sets spoken	Number of sentences spoken by males	Number of sentences spoken by females	Total
1	194	49	243
2	140	10	150
1 & 2	80	220	300
			693

In order to have a representative amount of data for training and testing purposes, the data was split into a 70% training-30% test ratio, maintaining the split based on the information in Table 4.1 as well. Any speaker who spoke both sentence sets will be found exclusively in either the training or test sets, not in both. Table 4.2 below describes the SUN Speech training and test sets used.

Table 4.2: SUN Speech training and test data

	Training sentences	**Test sentences**
Male speakers Set 1	134	60
Male speakers Set 2	100	40
Male speakers Set 1 & 2	60	20
Female speakers Set 1	40	9
Female speakers Set 2	10	0
Female speakers Set 1 & 2	140	80
TOTAL	**484 (69.8%)**	**209(30.2%)**

4.5.1. TIMIT data for adaptation and re-estimation

Recall that the adaptation data (for MLLR, MAP and re-estimation) used for the TIMIT-based recogniser is the Afrikaans SUN Speech training set. Since the quantity of TIMIT speech data vastly exceeds that of the available SUN Speech data, it would be improper to use the entire

TIMIT training set for the reverse case, i.e. when adapting the SUN Speech-based recogniser on TIMIT English data. Hence only a portion of the TIMIT training set was used for this purpose. The quantity of data selected for this adaptation was chosen so as to maintain the ratio between duration of source language data to duration of adaptation data.

This chapter described the experimental protocol that was used during the investigation. An overview of the methodology that was used during the study was presented. The tools that were used were also described. The general recogniser training methodology was then covered, including the configuration parameters within the Hidden Markov Toolkit. The performance criteria used during the course of the investigation to assess the recognisers' phoneme recognition abilities were then given. Finally, the test and training database particulars were described. The following chapter discusses the experiments that were conducted and their results.

Chapter 5

MAPPING EXPERIMENTS

This chapter details the experiments and the results that were found during the investigation. The main goal of this investigation is to be able to use the data of a source language, to train the initial acoustic models of a target language for which very little speech data may be available. To do this, an automatic technique for mapping the phonemes of the two data sets must be found. Using this technique, it would be possible to accelerate the development of a speech recognition system for a new language. This investigation has considered the English-to-Afrikaans phoneme mapping, as well as the Afrikaans-to-English phoneme mapping as well. In the latter case, for example, when Afrikaans phonemes are mapped to their nearest neighbour English phonemes, the initial acoustic models for the Afrikaans recogniser are obtained from the nearest-neighbour English model. The same reasoning applies for the English-to-Afrikaans phoneme mapping case as well.

This chapter is organised as follows. First, an overview of all the experiments that were carried out in this chapter is given. Initial experiments are carried out to establish baselines against which all subsequent experiments can be measured. The cross-language recognition abilities of the recognisers are then assessed. Finally, experiments are carried out to determine the effect of the number of MFCCs and states on the cross-language recognition results.

5.1. Overview of experiments in this Chapter

The sequence of experiments follows the order below:

- Baseline experiments are carried out in order to establish a yardstick by which all subsequent speech recognisers can be measured against. This set of experiments involved training and testing recognisers with same-language, same-database speech data in order to establish baseline performance figures.
- The cross-language capabilities of the respective speech recognisers on the mapped cross-language data are then assessed. The effect of the number of mixture components on the cross-language performance of the speech recognisers is then determined. The optimum number of mixture components, that yield the best cross-language phoneme recognition results, is determined for each distance measure.
- Next the impact of including the delta and acceleration MFCCs on the recogniser's cross-language recognition is determined. The optimum number of MFCCs is determined for each distance measure by evaluating the cross-language recognition results.
- The effect of the number of HMM states on cross-language phoneme recognition is then investigated. The impact on phoneme recognition of using the information from only the middle state (for three-state HMMs), as well as from all states is examined. These are evaluated by determining their impact on the cross-language recognition rates of the recognisers.

5.2. Baseline establishment experiments

In order to establish baseline measurements against which all subsequent experimental results can be measured, initial experiments were carried out to determine benchmark phoneme recognition performance figures for the English- and Afrikaans-trained recognisers.

5.2.1. Testing English data on English-trained recogniser

The English recogniser was trained using the TIMIT SI (phonetically-diverse) and SX (phonetically-compact) English training data and tested using the English TIMIT SI & SX test

data set. Single-state HMMs were used, and the number of mixtures per state was incremented (in steps of one) from one to four. Table 5.1 displays the performance of the TIMIT-based English recogniser on the TIMIT test set.

Table 5.1: Performance of TIMIT-based English recogniser on the TIMIT test set for mixture components from 1 to 4 using single-state HMMs

No. of mixtures	1	2	3	4
%correct	56.66	60.27	62.37	**63.71**
Accuracy	47.57	53.31	55.66	**57.97**

The fully trained English recogniser correctly recognised 63.71% of the English phoneme set with an accuracy of 57.97% when using four mixture components.

5.2.2. Testing Afrikaans data on Afrikaans-trained recogniser

The Afrikaans phoneme recogniser was trained using 70% of the available SUN Speech Afrikaans data. Here again, single-state HMMs were used and the number of mixtures used was increased from one to four. To obtain the benchmark performance figures, the Afrikaans recogniser was tested with the remaining 30% Afrikaans data. The results are shown in Table 5.2.

Table 5.2: Performance of SUN Speech-based Afrikaans recogniser on the Afrikaans test set for mixture components from 1 to 4 using single-state HMMs

No. of mixtures	1	2	3	4
%correct	49.78	56.63	59.03	**61.89**
Accuracy	35.27	42.83	45.59	**50.24**

The recogniser correctly identified 61.89% of the Afrikaans phonemes with an accuracy of 50.24%.

5.3. Effect of number of mixture components on mapping experiments

For this set of experiments, the number of MFCCs was maintained at 39 (i.e. both the delta and acceleration coefficients were included) and only single-state HMMs were considered. This part of the study investigated how the number of mixture components influences the SUN Speech to TIMIT mapping phoneme recognition results, as well as for the reverse case.

5.3.1. Effect of number of mixture components on cross-language recognition rate for Afrikaans to English phoneme mapping

Table 5.3 lists the phoneme recognition results per distance measure for mixture components from 1 to 4. Note as well that these results were obtained using the TIMIT-trained English recogniser using only the acoustic mapping technique; no adaptation had been performed at this stage on the English recogniser. The best results obtained per distance measure are shown in bold.

Table 5.3: Afrikaans to English phoneme mapping - Performance of English (TIMIT-based) recogniser on Afrikaans data per distance measure for mixture components from 1 to 4 using single-state HMMs

Distance Measure	No. of mixtures	1	2	3	4
KL	%correct	24.48	**26.55**	18.14	25.56
	Accuracy	6.21	**8.52**	2.46	7.87
BHA	%correct	24.25	**26.85**	18.27	24.42
	Accuracy	6.04	**8.78**	2.69	8.11
MAH	%correct	15.36	15.25	14.23	**18.78**
	Accuracy	-3.70	-0.20	-1.03	**1.29**
EUC	%correct	**28.42**	24.82	27.79	27.88
	Accuracy	8.21	7.23	8.51	**10.11**
L2	%correct	1.65	**19.91**	19.10	4.27
	Accuracy	-7.33	**12.89**	11.81	-2.59
JM	%correct	24.25	**26.85**	18.27	24.42
	Accuracy	6.04	**8.78**	2.69	8.11
Manual	%correct	26.93	27.27	**28.38**	27.55
	Accuracy	7.08	9.74	10.53	**11.59**

From Table 5.3, only the Mahalanobis distance measure shows progressive improvements in phoneme recognition performance up to four mixtures. The Euclidean distance metric achieves its highest recognition performance when just a single Gaussian is used. The other four distance metrics have their peak recognition performance when using two mixture components.

Overall, though, the Euclidean distance measure outperforms the other distance measures by 1.57% when using a single Gaussian. Second-best performance is identically delivered by the two-mixture Bhattacharyya and Jeffreys-Matusita measures. Recall that the Jeffreys-Matusita distance measure is derived from the Bhattacharyya distance measure, and for this set of experiments yielded identical mapping (and hence identical phoneme recognition) results.

The results shown in Table 5.3 are not comparable to the 63.71% correctly identified phonemes obtained when the Afrikaans recogniser was tested with the remaining 30% Afrikaans test data in the baseline establishment experiments. This 30% test data is a subset of the SUN Speech database, and is thus very similar to the training data. Moreover, the approach of training a new language recogniser from scratch with limited amounts of speech data available is not practical in a continuous speech recognition system. The primary purpose of this investigation was only to find the optimal distance measure for mapping phonemes to quickly generate initial acoustic models in a new language.

These results should be compared with the manual phonetic mapping procedure carried out by the phonetic expert that yielded a correctly recognised phoneme figure of 28.38% and an accuracy of 11.59%. From Table 5.3 (for the Afrikaans to English phoneme mapping experiments), it can be seen that all the distance metrics used, barring the Mahalanobis and L2 measures, had comparative performance to the manual mapping performed by the phonetic expert. In fact, when using 39 MFCCs, the Euclidean distance measure actually marginally outperformed the results achieved by the manual mapping process, although only by an improvement of 0.04% for the correctly identified phonemes.

According to the work done in [3], phonetically there are just two Afrikaans phoneme classes

in the SUN Speech database that do not appear in the English part of the database (these are represented by the [R] and [] phonemes or by their numerical ASCII codes of 82 and 94 respectively). These were grouped into a single [r] class during the manual mapping procedure. It should be noted that the recognition results for this phoneme model were the poorest for the manual mapping carried out by the phonetic expert (recognition rate = 14.9%), indicating that the manual mapping for these two phoneme classes is not a true indication of their acoustic nature.

5.3.2. Effect of number of mixture components on cross-language recognition rate for English to Afrikaans phoneme mapping

Table 5.4 lists the phoneme recognition results per distance measure for mixture components from 1 to 4 for the English to Afrikaans phoneme mapping case. Once again, these results are obtained using the SUN Speech-trained Afrikaans recogniser using only the acoustic mapping technique and no further adaptation.

Table 5.4: English to Afrikaans phoneme mapping - Performance of Afrikaans (SUN Speech-based) recogniser on English data per distance measure for mixture components from 1 to 4 using single-state HMMs

Distance Measure	No. of mixtures	1	2	3	4
KL	%correct	**33.70**	26.19	18.24	19.89
	Accuracy	**25.87**	18.65	13.08	13.98
BHA	%correct	**34.22**	26.57	18.39	22.47
	Accuracy	**26.25**	18.97	13.15	16.80
MAH	%correct	**20.41**	16.60	11.17	18.70
	Accuracy	**12.01**	9.51	7.21	11.00
EUC	%correct	**24.88**	21.08	16.01	15.91
	Accuracy	**18.39**	14.77	11.23	11.63
L2	%correct	6.89	6.84	6.89	**8.14**
	Accuracy	5.97	5.73	6.10	**6.82**
JM	%correct	**34.22**	26.57	18.39	22.47
	Accuracy	**26.25**	18.97	13.15	16.80

From Table 5.4, all the distance measures, used excluding the L2 metric, deliver optimum performance when using a single Gaussian. The L2 metric displays marginal improvement in phoneme recognition up to four mixture components. Overall, however, the L2 distance measure displays the poorest recognition performance when compared to the other measures.

Best performance for the English to Afrikaans phoneme mapping case is delivered by the single-Gaussian Bhattacharyya and Jeffreys-Matusita distance measures respectively which yield a phoneme recognition accuracy of 34.22% and an accuracy of 26.25%. Second-best performance is delivered by the single-Gaussian Kullback-Leibler distance measure (%correct = 33.70% and %accuracy = 25.87%).

5.4. Effect of excluding delta and acceleration MFCCs on cross-language phoneme recognition rate

For this set of experiments, the effect of including the delta and acceleration Mel frequency cepstral coefficients was investigated for both the SUN Speech-to-TIMIT mapping, as well as the TIMIT-to-SUN Speech case as well. Each of the six distance measures described in Chapter 3 was calculated for every TIMIT-SUN Speech phoneme pair (a 38-by-52 distance matrix was computed for each distance measure) and then for every SUN Speech-TIMIT phoneme pair (calculating a 52-by-38 distance matrix). The number of mixture components was varied from one to four, however only the strongest mixture component was used in the acoustic distance calculation.

5.4.1. Effect of number of MFCCs on cross-language recognition rate for Afrikaans to English phoneme mapping

In the first group of experiments, each SUN Speech phoneme was then mapped to the closest TIMIT phoneme (no distance threshold was applied), using 39, 26 and finally 13 MFCCs.

The results obtained using 39 MFCCs appears in Table 5.3 and will not be repeated in this section.

The mapped SUN Speech Afrikaans test data was then recognised by the trained TIMIT-based English recogniser. The phoneme recognition results when the delta-delta (acceleration) coefficients are excluded appear in Table 5.5 (the best results for a particular number of MFCCs in bold). The results when both the delta and delta-delta coefficients are excluded appear in Table 5.6.

Table 5.5: Afrikaans to English phoneme mapping - Performance of English (TIMIT-based) recogniser on mapped Afrikaans data per distance measure for single-state HMMs using mixture components from 1 to 4 and excluding the acceleration MFCC components

Distance Measure	No. of mixtures	1	2	3	4
KL	%correct	13.97	14.73	**15.47**	14.73
	Accuracy	2.73	4.37	4.45	**5.45**
BHA	%correct	13.88	**16.00**	15.41	15.25
	Accuracy	2.77	4.59	4.56	**5.40**
MAH	%correct	8.63	**14.24**	10.86	11.17
	Accuracy	0.84	**4.44**	2.54	3.11
EUC	%correct	17.56	16.89	**22.63**	15.60
	Accuracy	4.75	5.77	**8.40**	5.42
L2	%correct	**22.88**	17.67	12.97	2.92
	Accuracy	**16.26**	16.12	7.01	1.08
JM	%correct	13.88	**16.00**	15.41	15.25
	Accuracy	2.77	4.59	4.56	**5.40**

Table 5.6: Afrikaans to English phoneme mapping - Performance of English (TIMIT-based) recogniser on mapped Afrikaans data per distance measure for single-state HMMs using mixture components from 1 to 4 and excluding the acceleration & delta MFCC components

Distance Measure	No. of mixtures	1	2	3	4
KL	%correct	14.95	15.78	**16.02**	14.64
	Accuracy	9.02	11.60	**11.73**	10.74
BHA	%correct	14.09	14.90	**17.76**	14.58
	Accuracy	8.38	11.14	**12.82**	10.80
MAH	%correct	8.53	11.30	**15.21**	10.51
	Accuracy	5.37	8.83	**11.04**	8.22
EUC	%correct	17.63	13.99	**18.40**	15.99
	Accuracy	11.35	10.15	**14.11**	11.60
L2	%correct	**16.51**	14.76	14.34	5.84
	Accuracy	**14.36**	14.24	13.93	5.42
JM	%correct	14.09	14.90	**17.76**	14.58
	Accuracy	8.38	11.14	**12.82**	10.80

From Tables 5.5 and 5.6 (for the Afrikaans to English phoneme mapping experiments), it can be seen that the exclusion of both the delta and acceleration MFCCs produces better recognition results than when just the acceleration coefficients are excluded. All distance measures, excluding the L2 distance metric, exhibit improved phoneme recognition results when both the delta and acceleration coefficients are excluded. However, the best overall results are achieved when both the delta and acceleration coefficients are included, as shown in Table 5.3. The only exception to this is the L2 distance metric that delivers peak performance (%correct = 22.88% and %accuracy=16.26%) when both the delta and acceleration coefficients are excluded.

5.4.2. Effect of number of MFCCs on cross-language recognition rate for English to Afrikaans phoneme mapping

The same experiment as above was carried out, only in this case mapping from the English (TIMIT) phonemes to the Afrikaans (SUN Speech) phoneme set.

The results obtained using 39 MFCCs appears in Table 5.4 and will again not be repeated in this section.

The results for these experiments when the delta-delta (acceleration) coefficients are excluded are listed in Table 5.7 (once again, the best results for a particular number of MFCCs in bold). The results when both the delta and delta-delta coefficients are excluded appear in Table 5.8.

Table 5.7: English to Afrikaans phoneme mapping - Performance of Afrikaans (SUN Speech-based) recogniser on mapped English data per distance measure for single-state HMMs using mixture components from 1 to 4 and excluding the acceleration MFCC components

Distance Measure	No. of mixtures	1	2	3	4
KL	%correct	**14.67**	13.68	10.06	11.92
	Accuracy	**10.27**	9.14	7.25	8.25
BHA	%correct	**14.76**	12.27	10.83	11.89
	Accuracy	**10.36**	8.47	7.68	8.20
MAH	%correct	10.62	8.98	**12.18**	9.73
	Accuracy	8.48	6.78	**8.58**	6.90
EUC	%correct	**14.06**	13.56	11.56	9.68
	Accuracy	**9.95**	9.15	7.79	6.61
L2	%correct	5.41	**9.91**	5.57	3.21
	Accuracy	5.29	**9.25**	5.18	3.00
JM	%correct	**14.76**	12.27	10.83	11.89
	Accuracy	**10.36**	8.47	7.68	8.20

Table 5.8: English to Afrikaans phoneme mapping - Performance of Afrikaans (SUN Speech-based) recogniser on mapped English data per distance measure for single-state HMMs using mixture components from 1 to 4 and excluding the acceleration & delta MFCC components

Distance Measure	No. of mixtures	1	2	3	4
KL	%correct	8.88	**10.25**	6.12	9.79
	Accuracy	7.13	**8.06**	5.06	7.81
BHA	%correct	8.94	**10.83**	5.72	9.79
	Accuracy	7.20	**8.42**	4.76	7.82
MAH	%correct	8.01	**12.49**	8.37	8.77
	Accuracy	6.52	**10.54**	6.62	7.01
EUC	%correct	**12.00**	8.86	8.75	10.30
	Accuracy	**9.09**	7.08	6.94	8.24
L2	%correct	**6.72**	3.45	5.24	4.46
	Accuracy	**6.07**	3.01	4.83	4.41
JM	%correct	8.94	**10.83**	5.72	9.79
	Accuracy	7.20	**8.42**	4.76	7.82

From Tables 5.7 & 5.8, it is evident that excluding both the delta and acceleration MFCCs leads to significantly poorer recognition results for all distance measures. The peak performance recognition results appear in Table 5.4. For the L2 distance measure, the best results are achieved when only the acceleration MFCCs are excluded. When comparing the results where only the delta and where both the delta and acceleration coefficients are excluded, for the English to Afrikaans phoneme mapping case, better results are obtained when just the delta MFCCs are excluded. This indicates that there is useful phoneme recognition information present in the acceleration coefficients as well.

5.5. Effect of number of states on cross-language recognition rate

For this set of experiments, the number of MFCCs was maintained at 39 (i.e. both the delta and acceleration coefficients were included). This part of the study looked at how the number of states influenced the SUN Speech to TIMIT, and the TIMIT to SUN Speech mapping

phoneme recognition results. Note that in HTK, single-state HMMs are modelled as three-state HMMs. This is done to cater for the non-emitting ENTER and EXIT states defined in HTK. For the first set of experiments, only the middle state was used in calculating the distance metrics. Although the number of mixture components was varied from 1 to 4, only the strongest mixture component was used in the distance metric calculation.

5.5.1. Effect of number of states on recognition rate for Afrikaans to English phoneme mapping

Table 5.9 lists the phoneme recognition results per distance measure for mixture components from 1 to 4 when 3 states are used. Note as well that these results were obtained using the TIMIT-trained English recogniser using only the acoustic mapping technique; no adaptation had been performed at this stage on the English recogniser.

Table 5.9: Afrikaans to English phoneme mapping - Performance of English (TIMIT-based) recogniser on Afrikaans data per distance measure for mixture components from 1 to 4 using 3 HMM states

Distance Measure	No. of mixtures	1	2	3	4
KL	%correct	14.97	**22.45**	12.64	17.24
	Accuracy	1.99	**5.44**	-0.90	2.00
BHA	%correct	15.90	**18.82**	13.90	18.31
	Accuracy	2.24	**3.87**	-0.09	2.24
MAH	%correct	16.30	**20.20**	13.01	19.03
	Accuracy	2.42	**4.68**	0.22	2.88
EUC	%correct	18.69	**21.93**	17.97	16.70
	Accuracy	2.82	**4.63**	1.54	2.51
L2	%correct	6.90	5.25	8.00	**19.17**
	Accuracy	0.81	-0.55	1.15	**6.48**
JM	%correct	15.90	**18.82**	13.90	18.31
	Accuracy	2.24	**3.87**	-0.09	2.24

As before, it is evident from Table 5.9 that the best cross-language phoneme recognition

performance is achieved when two mixture components are used. The only distance measure that bears an exception to this is the L2 metric that has peak recognition performance for four mixture components.

Overall, though, the Kullback-Leibler distance measure outperforms the other distance measure by at least 0.52% when using just two mixture components. Second-best performance is delivered by the two-mixture Euclidean distance measure.

This can now be compared to the results obtained using single-state HMMs, listed in Table 5.3. When comparing the results from Table 5.3 with the results for 3-state HMMs in Table 5.9, it can be seen using single-state HMMs generally produce better results for all distance measures. The only exception is the Mahalanobis distance metric that displays better phoneme recognition accuracy when 3 states are used.

5.5.2. Effect of number of states on cross-language recognition rate for English to Afrikaans phoneme mapping

Table 5.10 lists the phoneme recognition results per distance measure for mixture components from 1 to 4 using 3 states for the English to Afrikaans phoneme mapping case. Note as well that these results were obtained using the SUN Speech-trained Afrikaans recogniser using only the acoustic mapping technique; no adaptation had been performed at this stage on the Afrikaans recogniser.

Table 5.10: English to Afrikaans phoneme mapping - Performance of Afrikaans (SUN Speech-based) recogniser on English data per distance measure for mixture components from 1 to 4 using 3 HMM states

Distance Measure	No. of mixtures	1	2	3	4
KL	%correct	9.08	11.15	**12.61**	7.15
	Accuracy	7.24	8.82	**9.67**	5.83
BHA	%correct	9.62	10.58	**12.69**	7.82
	Accuracy	7.46	8.35	**9.80**	6.20
MAH	%correct	15.07	11.90	11.41	**13.07**
	Accuracy	**11.18**	9.35	8.52	9.14
EUC	%correct	10.59	13.05	**13.61**	11.99
	Accuracy	8.21	9.35	**9.91**	8.46
L2	%correct	5.00	**12.10**	4.75	4.04
	Accuracy	4.86	**11.73**	4.58	3.90
JM	%correct	9.62	10.58	**12.69**	7.82
	Accuracy	7.46	8.35	**9.80**	6.20

From Table 5.10, four of the distance measures (the Kullback-Leibler, Bhattacharyya, Euclidean and Jeffreys-Matusita) show progressive improvements in phoneme recognition performance up to three mixtures. The Mahalanobis distance measure has its peak recognition performance for four mixture components, while the L2 metric exhibits its best results using two mixture components only.

Overall, though, the Euclidean distance measure outperforms the other distance measure by at least 0.92% when using just three mixture components. Second-best performance is delivered by the three-mixture Jeffreys-Matusita and Bhattacharyya measures.

These results using three HMM states can now be compared to the results obtained using single-state HMMs, listed in Table 5.4. When comparing the results from Table 5.4 with the results for 3-state HMMs in Table 5.10, it can be seen that using single-state HMMs generally produce better results for all distance measures, as was found in the reverse Afrikaans to

English phoneme mapping case as well. The only exception is the L2 distance metric that displays improved phoneme recognition (%correct and %accuracy) when 3 states are used.

5.6. Effect on cross-language recognition rate of using all states in distance metric calculation

For this set of experiments, the number of MFCCs was maintained at 39 (i.e. both the delta and acceleration coefficients were included). This part of the study looked at how using all the states in the distance metric calculation influenced the SUN Speech to TIMIT, and the TIMIT to SUN Speech mapping phoneme recognition results. Although the number of mixture components was varied from 1 to 4, only the strongest mixture component per state was used in the distance metric calculation. The difference between this set of experiments compared to the previous one, is that all states were used in calculating the distance metrics, i.e. each of the distance metrics were calculated for the strongest mixture component per state and added for a total distance measure per distance measure.

5.6.1. Effect of using all states on cross-language recognition rate for Afrikaans to English phoneme mapping

Table 5.11 lists the phoneme recognition results per distance measure for mixture components from 1 to 4 when all 3 states are used in the phone distance calculation. As before, these results were obtained using the TIMIT-trained English recogniser using only the acoustic mapping technique, and excluding any further adaptation.

Table 5.11: Afrikaans to English phoneme mapping - Performance of English (TIMIT-based) recogniser on Afrikaans data per distance measure for mixture components from 1 to 4 using all 3 HMM states in distance calculation

Distance Measure	No. of mixtures	1	2	3	4
KL	%correct	16.49	**22.53**	16.67	16.94
	Accuracy	2.44	**5.44**	2.78	1.75
BHA	%correct	18.21	**21.87**	18.22	19.51
	Accuracy	3.59	**6.23**	3.50	2.61
MAH	%correct	15.69	**20.43**	18.63	17.97
	Accuracy	3.91	**5.39**	2.37	1.91
EUC	%correct	19.37	19.11	**22.67**	19.82
	Accuracy	4.18	3.52	**4.90**	2.86
L2	%correct	**16.85**	5.52	8.28	9.42
	Accuracy	**13.09**	0.22	-0.60	-0.01
JM	%correct	17.83	19.62	19.59	**20.67**
	Accuracy	**3.41**	3.13	2.22	2.88

It is evident from Table 5.11, that the best cross-language phoneme recognition performance is achieved when two mixture components are used. The Euclidean, L2 and Jeffreys-Matusita distance measures exhibit peak performance when using three, one and four mixture components respectively.

The three-mixture Euclidean distance marginally outperforms the Kullback-Leibler distance measure by 0.14%. In terms of accuracy, the L2 metric using a single Gaussian delivers the best phoneme recognition by 6.86%.

It should be noted that this set of experiments is the only one in which the Bhattacharyya and Jeffreys-Matusita distance measures give different results.

These results can now be compared to the results obtained using single-state HMMs that are

listed in Table 5.3, and those using the middle state for 3-state HMMs in Table 5.9. When comparing the results obtained when only the middle state of the 3-state HMMs is used, improved results are obtained when using all 3 states in the distance calculation for all distance measures except the L2 metric. However, when compared to the results obtained when using single-state HMMs (Table 5.3), using all 3 HMM states in the distance calculation generally produces worse results for all distance measures. The only exception is the Mahalanobis distance metric that displays improved phoneme recognition accuracy when all 3 states are used in the distance calculation.

5.6.2. Effect of using all states on cross-language recognition rate for English to Afrikaans phoneme mapping

Table 5.12 lists the phoneme recognition results per distance measure for mixture components from 1 to 4 when all 3 states are used in the phone distance calculation. Again it should be noted that these results were obtained using the SUN Speech-trained Afrikaans recogniser using only the acoustic mapping technique; no adaptation had been performed at this stage on the Afrikaans recogniser.

Table 5.12: English to Afrikaans phoneme mapping - Performance of SUN Speech-based recogniser on English data per distance measure for mixture components from 1 to 4 using all 3 HMM states in distance calculation

Distance Measure	No. of mixtures	1	2	3	4
KL	%correct	**12.01**	11.58	9.99	9.44
	Accuracy	8.70	**9.29**	7.91	7.01
BHA	%correct	**11.64**	11.38	10.11	9.67
	Accuracy	8.51	**9.08**	7.96	7.25
MAH	%correct	9.75	**14.19**	9.76	8.54
	Accuracy	7.18	**11.22**	7.43	6.27
EUC	%correct	10.12	10.85	**14.42**	9.58
	Accuracy	7.86	8.66	**10.67**	7.24
L2	%correct	4.41	5.11	9.11	**9.74**
	Accuracy	4.04	4.87	7.80	**7.88**
JM	%correct	11.42	**13.01**	11.53	8.98
	Accuracy	8.88	**9.52**	8.84	6.91

From Table 5.12, overall best performance for the English to Afrikaans mapping case is achieved by the Euclidean distance measure (% correct = 14.42% and accuracy of 10.67%). This is followed by the Mahalanobis measure that achieves a better accuracy figure of 11.22%.

As in the reverse phoneme mapping experiments listed in the previous section, the Bhattacharyya and Jeffreys-Matusita distance measures produce different recognition results. This is due to the technique of adding the distance contributions per state to determine an overall distance figure used to determine which phoneme pairs are the closest.

This can now be compared to the results obtained using single-state HMMs that are listed in Table 5.4, and those using the middle state for 3-state HMMs in Table 5.10. When comparing the results obtained when only the middle state of the 3-state HMMs is used, improved results are only obtained when using all 3 states in the distance calculation for the Mahalanobis and Euclidean metrics. However, when compared to the results obtained when using single-state HMMs (Table 5.4), using all 3 HMM states in the distance calculation generally produces worse results for all distance measures. The only exception is the L2 distance metric that displays improved phoneme recognition.

5.7. Summary of Chapter findings

Baseline establishment experiments were conducted to determine benchmark phoneme recognition performance figures. The English recogniser when trained and tested with the English TIMIT speech data correctly recognised 63.71% of the English phoneme set with an accuracy of 57.97%. The Afrikaans recogniser when trained and tested with the SUN Speech Afrikaans data correctly recognised 61.89% of the Afrikaans phonemes with an accuracy of 50.24%.

It was found that the recogniser's cross-language performance increases up to a limit as the number of Gaussian mixtures increase. This was found for both the English-to-Afrikaans and Afrikaans-to-English cases. The optimum number of Gaussian mixtures differs for each of the distance measures.

It was also found that the inclusion of the delta and acceleration MFCCs improve the mapping and the recognition system's cross-language phoneme recognition rate. This would indicate that there is useful information in the delta and acceleration components that aids in distinction between the phonemes.

The results of the experiments showed that single-state HMMs deliver the optimum cross-language phoneme recognition results. Although the number of mixture components was varied from 1 to 4, only the strongest mixture component was used in the distance metric calculation. Note that in HTK, single-state HMMs are modelled as three-state HMMs. This is done to cater for the non-emitting ENTER and EXIT states defined in HTK. For the first set of experiments, only single-state HMMS were used. For the second set of experiments, only the middle state (of the three states) was used in calculating the distance metrics, and for the third set of experiments all three states were used in the distance calculation. The latter two experiments produced poorer cross-language recognition results than when only a single-state HMM is used. Using the Bhattacharyya and Jeffreys-Matusita distance measures in phoneme mapping resulted in the best cross-language phoneme recognition rates.

These experiments have also demonstrated that the choice of acoustic distance measure for the mapping does influence the results obtained. Generally, the Bhattacharyya, Jeffreys-Matusita, Euclidean and Kullback-Leibler distance measures perform the best when mapping the phonemes to a target language, be this target language either English or Afrikaans. The relatively simple Euclidean metric exhibits good results and has shown that it is able to map the acoustic space of phoneme models relatively well. The L2 and Mahalanobis distance measures, although useful in other pattern recognition applications, do not appear to be suitable for determining the degree of acoustic similarity between two phoneme classes. They have generally performed poorly in comparison with the other distance metrics. A suggestion for the difference in cross-language phoneme recognition performance is due to the linearity of each of the distance measures. The distance measures that generally performed better were more linear and produced more consistently comparable distances between the phoneme classes. The measures that performed poorly, generally exhibited either very small or very

large distance results between the phonemes, with the result that there was a degree of uncertainty about the whether the closest phoneme chosen was in fact a true indication of its closeness.

Now that the initial baselines have been established and the optimum number of MFCCs for the cross-language phoneme recognition results has been determined, the next step is to adapt the available data. The next chapter investigates the effect of applying MLLR and MAP adaptations on the initial acoustic models.

Chapter 6

SPEAKER ADAPTATION EXPERIMENTS

This part of the investigation looks at performing MLLR adaptation and then MAP adaptation on the recogniser models (both the mean and variance) to adapt them closer to the target language data. Nieuwoudt [3, 8] successfully demonstrated this technique for same-language but different databases, and for cross-language applications as well.

This chapter is organised as follows:

- The first experiment in this chapter looks at performing MLLR and then MAP adaptations (using mapped cross-language data) on the recogniser models (both the mean and variance) to adapt them closer to the target language data. The cross-language phoneme models that were previously found to produce the best recognition results were used for initial models, with the nearest-neighbour phonemes providing the adaptation data. This is illustrated for the Afrikaans to English phoneme mapping case in Figure 6.1 below:

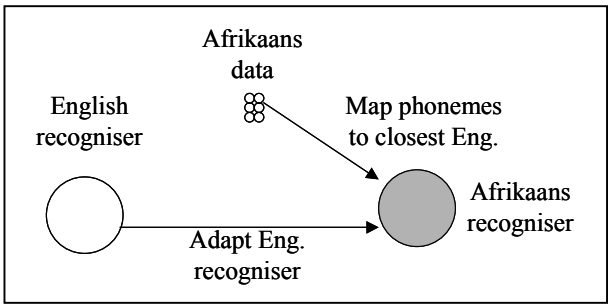

Figure 6.1: Illustration of data mapping and recogniser adaptation technique

- Embedded Baum-Welch re-estimation (EBWR) is then performed on the adapted models to improve their cross-language recognition abilities. Once again the mapped data, provided by the nearest neighbour experiments that provided the best recognition results, is used for the re-estimation.

The first step is to investigate the effect of adapting the Afrikaans data.

6.1. Effect of performing MLLR and MAP adaptations using Afrikaans to English mapped data

For the SUN Speech to TIMIT mapping MLLR and MAP adaptation, the adaptation data that is used is the Afrikaans SUN Speech training set. The effect of the adaptation is quantified on the SUN Speech test training set. All experiments are carried out using single-state HMMs.

As an interesting aside, the adapted model set is also tested on the TIMIT English test data to evaluate the "bilingual" capabilities of the adapted recogniser, as well as to determine how much degradation in performance the adaptation has brought about. In Table 6.1, results are shown when the "best" number of mixture components is used, as has been determined in previous experiment sets. The best results are shown in bold. Table 6.1 also includes the results for the Manual phoneme mapping process carried out by the phonetic expert.

Table 6.1: Afrikaans to English phoneme mapping - Performance of English (TIMIT-based) recogniser on Afrikaans data per distance measure after MLLR and MAP adaptation

Distance Measure		SUN Test Pre-adaptation	SUN test after MLLR & MAP – "best" mix.	TIMIT test after MLLR & MAP
KL	%correct	26.55	36.56	46.15
	Accuracy	8.52	16.98	38.81
BHA	%correct	26.85	**40.01**	44.96
	Accuracy	8.78	**21.57**	38.40
MAH	%correct	18.78	20.63	44.30
	Accuracy	1.29	7.79	39.90
EUC	%correct	28.42	31.76	49.26
	Accuracy	8.21	11.77	40.61
L2	%correct	19.91	13.41	14.06
	Accuracy	12.89	13.41	13.86
JM	%correct	26.85	**40.01**	44.96
	Accuracy	8.78	**21.57**	38.40
Manual mapping	%correct	28.38	37.95	**49.59**
	Accuracy	10.53	19.08	**44.16**

There are a number of observations from Table 6.1. Firstly, note that the Bhattacharyya and Jeffreys-Matusita distance measures after MLLR and MAP adaptation outperform all other distance measures, including the manual mapping technique, with a phoneme recognition rate of 40.01% (accuracy of 21.57%). This is an improvement in recognition rate of 13.16%, and a gain in accuracy of 12.79% over the unadapted recogniser. In terms of recognition performance, the manual mapping technique undertaken by the phonetic expert produces the next-best phoneme recognition results (%correct of 37.95% and 19.08%).

In terms of the "bilingual" nature of this recogniser, it should be borne in mind that the fully trained English recogniser correctly recognised 63.71% of the English phoneme set (see Table 5.1) with an accuracy of 57.97%. After adaptation, the adapted recogniser is tested on the TIMIT test set. For the Jeffreys-Matusita and Bhattacharyya-based recognisers, this recognition performance on the English phoneme set has dropped to 44.96%, with an accuracy of 38.40%. This represents degradation in phoneme recognition performance of 18.75% and in

accuracy of 19.57%.

6.2. Effect of performing MLLR and MAP adaptations using English to Afrikaans mapped data

For the TIMIT to SUN Speech mapping MLLR and MAP adaptation, the adaptation data that is used is a sample of the TIMIT SI and SX training set. Approximately the same duration of adaptation data that is used for the SUN Speech to TIMIT mapping case is used for this case as well. This is done so that the adaptation process is not unduly biased by the quantity of adaptation data available, as would have been the case if the entire TIMIT training set had been used. The effect of the adaptation is quantified on the entire TIMIT SI & SX test training set. As before, the adapted model set is also tested on the SUN Speech Afrikaans test data to evaluate the "bilingual" capabilities of the adapted recogniser, as well as to determine how much degradation in performance the adaptation has brought about.

Table 6.2: English to Afrikaans phoneme mapping - Performance of Afrikaans (SUN Speech-based) recogniser on English data per distance measure after MLLR and MAP adaptation

Distance Measure		TIMIT test pre-adaptation	TIMIT test after MLLR & MAP	SUN test after MLLR & MAP
KL	%correct	33.70	41.76	42.34
	Accuracy	25.87	**32.70**	28.26
BHA	%correct	34.22	**41.87**	41.94
	Accuracy	26.25	32.63	27.55
MAH	%correct	20.41	21.98	29.51
	Accuracy	12.01	15.52	19.17
EUC	%correct	24.88	32.48	**56.04**
	Accuracy	18.39	29.61	**50.82**
L2	%correct	8.14	15.57	17.78
	Accuracy	6.82	15.21	16.20
JM	%correct	34.22	**41.87**	41.94
	Accuracy	26.25	32.63	27.55

Once again, the best phoneme recognition results are obtained using the Jeffreys-Matusita and Bhattacharyya-transformed models (41.87% correctly recognised phonemes with an accuracy

of 32.63%). This represents a marginal 0.11% better recognition rate and a 0.07% worse accuracy rate than the next best model, the transformed Kullback-Leibler distance model. Moreover, this also represents a 7.65% improvement in phoneme recognition and a gain of 6.38% in accuracy over the untransformed model.

As far as the bilingual capabilities of this adapted SUN Speech recogniser go, this must be compared to the baseline figures achieved when training with the SUN Speech Afrikaans training set and tested with the SUN Speech test data. The recogniser correctly identified 61.89% of the Afrikaans phonemes with an accuracy of 50.24% (see Table 5.2). The best "bilingual" model is the adapted Euclidean model, which demonstrates a recognition rate on the Afrikaans data test set of 56.04% and an accuracy figure of 50.82%, down by 5.85% and up by 0.58% in phoneme recognition and accuracy respectively. This indicates satisfactory bilingual capabilities.

6.3. Effect on recognition rate of applying EBWR to models transformed using Afrikaans data

Single-pass embedded Baum-Welch re-estimation (EBWR) is performed on the MAP and MLLR transformed models. EBWR is performed in three iterations. As mentioned previously, EBWR uses all the adaptation data and adapts all the models concurrently.

Table 6.3: Afrikaans to English phoneme mapping - Performance of Re-estimated TIMIT-based recogniser on Afrikaans data per distance measure after Embedded Baum-Welch Re-estimation

Distance Measure		SUN test EBWR1	TIMIT test EBWR1	SUN test EBWR2	TIMIT test EBWR2	SUN test EBWR3	TIMIT test EBWR3
KL	%correct	40.13	43.79	41.46	42.60	42.37	42.02
	Accuracy	20.76	34.90	22.15	32.87	22.56	**32.10**
BHA	%correct	42.85	42.28	44.49	41.25	44.80	40.59
	Accuracy	25.16	33.85	26.68	31.89	26.91	30.98
MAH	%correct	23.74	34.77	26.91	32.37	29.31	31.40
	Accuracy	13.77	29.88	16.98	26.95	18.44	25.59
EUC	%correct	37.57	44.61	37.96	42.96	38.11	**42.27**
	Accuracy	13.78	32.78	13.13	30.77	12.45	29.52
L2	%correct	8.98	10.89	9.15	11.53	9.25	11.66
	Accuracy	8.98	10.81	9.15	11.43	9.25	11.56
JM	%correct	42.85	42.28	44.49	41.25	44.80	40.59
	Accuracy	25.16	33.85	26.68	31.89	26.91	30.98
Manual mapping	%correct	43.72	40.96	46.31	38.86	**47.93**	38.11
	Accuracy	24.81	35.03	26.51	32.16	**27.62**	31.31

From Table 6.3, it is clear how the performance of the TIMIT-based transformed recogniser improves for the SUN Speech Afrikaans test data with each EBWR iteration. The re-estimated model derived from the manual mapping technique outperforms the rest, with a recognition rate of 47.93% and an accuracy of 27.62%. This should now be compared with the results obtained when the SUN Speech recogniser is trained on SUN Speech data and tested with SUN Speech data. This had a phoneme recognition rate of 61.89%. The 13.96% degradation in performance should be put into context of the amount of effort and time that has been spared in getting the recogniser up to this level.

As far as the distance measures go, the Bhattacharyya and Jeffreys-Matusita metrics performed the next best, with recognition rates of 44.80% and accuracies of 26.91%.

Note that after the third iteration of EBWR, the recognition rate of the TIMIT-based recogniser for TIMIT English phonemes has dropped dramatically. The Euclidean-based

transformed and re-estimated model displays the best TIMIT English phoneme recognition rate of 42.27% with an accuracy of 29.52%.

Tables 6.4 and 6.5 list the phoneme classes with the best and worst recognition percentages per distance measure when using 39 MFCCs and after the final embedded Baum-Welch re-estimation.

Table 6.4: Top 5 recognition performances listed as a percentage of correctly recognised phonemes per distance measure for the SUN Speech to TIMIT mapping after 3 iterations of embedded Baum-Welch re-estimation

Distance Measure	**KL**	**BHA**	**MAH**	**EUC**	**L2***	**JM**	**Manual**
phoneme	cl	cl	cl	s	cl	cl	ch
%correct	91.8	95.9	95.9	86.1	44.7	95.9	94.8
phoneme	ae	z	ae	n		z	ay
%correct	80.9	79.9	73.3	84.3		79.9	89.5
phoneme	uw	ae	ay	cl		ae	cl
%correct	77.8	77.9	72.6	80.6		77.9	86.3
phoneme	z	f	aw	sh		f	k
%correct	76.2	74.1	72.1	78.4		74.1	83.1
phoneme	sh	aa	dh	ae		aa	oy
%correct	75.7	73.5	71.7	76.2		73.5	83.1

* For the L2 metric, the [cl] phoneme was the closest phoneme for all SUN Speech phonemes

Table 6.5: Bottom 5 recognition performances listed as a percentage of correctly recognised phonemes per distance measure for the SUN Speech to TIMIT mapping after 3 iterations of embedded Baum-Welch re-estimation

Distance Measure	**KL**	**BHA**	**MAH**	**EUC**	**L2***	**JM**	**Manual**
phoneme	uh	uh	z	uh	cl	uh	d
%correct	14.6	11.3	9.1	9.9	44.7	11.3	23.5
phoneme	g	dx	oy	b		dx	g
%correct	15.3	26.6	10.9	10.0		26.6	24.6
phoneme	d	d	r	oy		d	ah
%correct	24.0	26.7	12.8	11.6		26.7	26.6
phoneme	m	th	jh	m		th	uw
%correct	28.0	31.2	18.9	21.1		31.2	28.1
phoneme	th	m	eh	ch		m	dx
%correct	31.5	33.0	27.1	22.2		33.0	28.3

* For the L2 metric, the [cl] phoneme was the closest phoneme for all SUN Speech phonemes

From Table 6.4, it is evident that the [s], [sh], [f], [ae] and [cl] (silence) models were recognised the best. In general, the fricative sounds were recognised the best by the MLLR and MAP transformed EBWR-based recognisers. It should also be noted that while five of the six distance-based recognised exhibited excellent recognition results for the [ae] phoneme, the manual-mapped procedure had average phoneme recognition results for [ae].

Generally the [th], [dx], [uh], [d] and [m] displayed the poorest recognition results. Overall, the "stop" class of phoneme models tended to have the worst recognition results.

Table 6.6: Confusion matrix information for Afrikaans to English phoneme mapping

	Most commonly confused phoneme classes per distance measure					
Phoneme class	**KL**	**BHA**	**MAH**	**EUC**	**L2**	**JM**
th	t, cl, ih	uh, cl, t	cl, s	t, cl, dh		uh, cl, t
dx	cl, ih	cl, n		cl, ah, t		cl, n
uh	ih, ng, cl	cl, ng		cl, ah, ng		cl, ng
d	t, v, n	dh, n	n			dh, n
m	ih, cl	ih, n	cl, n	n, cl		ih, n

Table 6.6 lists the most commonly confused phoneme classes per distance measure for the set of worst recognised phonemes. From Table 6.6, it is evident that quite often, the greatest source of confusion is the [cl] or "silence" model. This is more than likely due to the acoustic differences between the two different speech databases used in the investigation. Another interesting observation is that the fricative [th] sound is often confused with the stop [t] sound. It can also be seen from Table 6.6 that the nasal [m] phoneme is often confused with the nasal [n] phoneme, something that should be expected.

6.4. Effect on recognition rate of applying EBWR to models transformed using English data

For the TIMIT to SUN Speech (English to Afrikaans) mapping case, single-pass embedded Baum-Welch re-estimation (EBWR) is performed on the MAP and MLLR transformed models. As before, EBWR is performed in three iterations.

Table 6.7: English to Afrikaans phoneme mapping - Performance of Re-estimated SUN Speech-based recogniser on English data per distance measure after Embedded Baum-Welch Re-estimation

Distance Measure		TIMIT test EBWR1	SUN test EBWR1	TIMIT test EBWR2	SUN test EBWR2	TIMIT test EBWR3	SUN test EBWR3
KL	%correct	45.15	40.87	45.48	40.43	**45.51**	40.39
	Accuracy	35.22	25.09	35.19	23.73	34.77	23.17
BHA	%correct	45.01	39.93	45.48	39.42	45.46	39.56
	Accuracy	34.81	24.19	34.85	22.88	34.70	21.99
MAH	%correct	25.70	27.15	28.87	23.59	31.73	24.00
	Accuracy	18.01	15.61	20.92	12.10	23.24	10.07
EUC	%correct	39.16	50.65	40.67	49.70	41.43	**49.30**
	Accuracy	35.75	44.78	37.03	43.42	**37.74**	**42.80**
L2	%correct	13.92	13.82	14.03	12.76	13.96	11.87
	Accuracy	13.65	12.82	13.79	11.85	13.70	11.09
JM	%correct	45.01	39.93	45.48	39.42	45.46	39.56
	Accuracy	34.81	24.19	34.85	22.88	34.70	21.99

From Table 6.7, it can be seen how the performance of the SUN Speech-based transformed

recogniser improves for the TIMIT English test data with each EBWR iteration. For this reverse mapping exercise from English to Afrikaans, the Kullback-Leibler based transformed re-estimated model outperforms the rest, with a recognition rate of 45.51% and an accuracy of 34.77%.. It should once again be borne in mind that only a portion of the TIMIT training data was used for the adaptation and re-estimation process. This can now be compared with the results obtained when the TIMIT recogniser is trained and tested with TIMIT data, which yielded a phoneme recognition rate of 63.71%. The adapted and re-estimated recogniser using English mapped data exhibits a considerable 18.20% degradation in phoneme recognition. Recall that the SUN Speech recogniser was trained on a limited amount of training data to begin with. This seems to indicate that the adaptation and re-estimation strategy suggested in this dissertation works better for recognisers that are trained with a large amount of data in the source language.

Note that after the third iteration of EBWR, the recognition rate of the SUN Speech-based recogniser for SUN Speech Afrikaans phonemes has dropped marginally only, and not as markedly as in the reverse-mapping case. The Euclidean-based transformed and re-estimated model displays the best SUN Speech Afrikaans phoneme recognition rate (49.30%) with an accuracy of 42.80%.

Table 6.8 and 6.9 list the phoneme classes with the best and worst recognition percentages per distance measure when using 39 MFCCs and after three iterations of embedded Baum-Welch re-estimation.

Table 6.8: Top 5 recognition performances listed as a percentage of correctly recognised phonemes per distance measure for the TIMIT to SUN Speech mapping after 3 embedded Baum-Welch re-estimation iterations

Distance Measure	KL	BHA	MAH	EUC	L2	JM
phoneme	s	s	n	s	s	s
%correct	89.9	89.9	68.1	92.7	99.5	89.9
phoneme	f	f	u	eh	p	f
%correct	84	83	62.8	75.2	98.7	83
phoneme	a	a	axi	t	sil	a
%correct	75.9	75.8	62.5	73.4	64.5	75.8
phoneme	n	n	oey	k		n
%correct	71	70.3	57.6	71.7		70.3
phoneme	eh	eh	ax	n		eh
%correct	68.9	68.5	55.6	71.5		68.5

Table 6.9: Bottom 5 recognition performances listed as a percentage of correctly recognised phonemes per distance measure for the TIMIT to SUN Speech mapping after 3 embedded Baum-Welch re-estimation iterations

Distance Measure	KL	BHA	MAH	EUC	L2	JM
phoneme	oey	oey	ao	r2	jh	oey
%correct	14	11.9	17.4	0.6	2.7	11.9
phoneme	q	zh	q	oi	b	zh
%correct	15.3	20.1	18.3	1.7	23.7	20.1
phoneme	ts	q	m	zh		q
%correct	17.8	20.3	19.1	22.3		20.3
phoneme	zh	ts	j	oey		ts
%correct	19.6	21.1	21.7	25.1		21.1
phoneme	u	g	p	sq		g
%correct	27.7	30.2	27.1	25.7		30.2

Firstly, note from the above two tables that only the best three recognised phoneme classes and the worst two recognised phoneme classes are provided for the L2 based recogniser. This is because the mapped SUN Speech phonemes were mapped to only five TIMIT phoneme classes during the mapping exercise.

From Table 6.8, it is clear that the [s], [f] and [n] models were recognised the best, as they were in the SUN Speech to TIMIT phoneme mapping case. In general, the fricative sounds were recognised the best by the MLLR and MAP transformed EBWR-based recognisers. It should also be noted that while for the reverse mapping case (mapping from Afrikaans to English case) the [cl] or "silence" model was one of the best-recognised phoneme classes, the same is not the case for the English phoneme to Afrikaans [sil] phoneme case where it exhibits average recognition results.

It can be seen from Table 6.9 that the [oey], [zh] and [q] displayed the poorest recognition results.

Table 6.10: Confusion matrix information for English to Afrikaans phoneme mapping

	Most commonly confused phoneme classes per distance Measure					
Phoneme class	**KL**	**BHA**	**MAH**	**EUC**	**L2***	**JM**
oey	ax, axi	ax, axi,eh	ax, axi	ax, axi		ax, axi,eh
zh	t, s	t, s		ch, t		t, s
q	d, x, m	d, k	l, ax			d, k

* For the L2 metric, the [cl] phoneme was the closest phoneme for all SUN Speech phonemes

Table 6.10 lists the most commonly confused phoneme classes per distance measure for the set of worst recognised phonemes. The diphthong [oey] phoneme (as in the word "b**ai**t") is most often confused with the vowel [ax] (as in "deb**i**t") and diphthong [axi] (as in the word "f**a**te") phoneme classes. The confusion between the like-sounding diphthong classes is to be expected. It can also be noted from Table 6.10 that the fricative [zh] phoneme (as in the word "a**z**ure") is often misclassified as the fricative [s] sound (as in "**s**ea"), or the stop [t] sound.

6.5. Summary of Chapter findings

This chapter has investigated the effect of adapting the available speech data using MLLR and MAP adaptation, and then applying EBWR to the adapted models. Although a combination of MLLR and MAP techniques have been used previously in speech adaptation studies, the combination of MLLR, MAP and EBWR in cross-language speech recognition is a unique contribution of this study.

The first set of experiments in this chapter looked at performing MLLR and then MAP adaptation on the Afrikaans to English mapped data. The Bhattacharyya and Jeffreys-Matusita distance measures after MLLR and MAP adaptation outperform all other distance measures, including the manual mapping technique. To test the "bilingual" nature of this recogniser after adaptation, the adapted recogniser is tested on the English TIMIT test set. As expected, there is a degradation in English (TIMIT) phoneme recognition performance after adaptation using the mapped Afrikaans data.

The second set of experiments investigates the effect of performing MLLR and then MAP adaptation on the English to Afrikaans mapped data. Once again, the best phoneme recognition results are obtained using the Jeffreys-Matusita and Bhattacharyya-transformed models. As far as the bilingual capabilities of this adapted SUN Speech recogniser go, the Afrikaans recogniser after adaptation did exhibit satisfactory bilingual capabilities, with the expected degradation in Afrikaans phoneme recognition.

The final set of experiments in this chapter looked at performing Embedded Baum-Welch Re-estimation on each of the English and Afrikaans recognisers after adaptation. This technique is found to further improve the cross-language recognition performance of the recognisers. Once again, the recognisers that were adapted with data that was mapped using the Bhattacharyya and Jeffreys-Matusita distance metrics performed the best. As expected, the bilingual recognition performance of the recognisers dropped with each successive EBWR iteration.

For the English recogniser adapted with Afrikaans speech data, the fricative sounds were recognised the best by the MLLR and MAP transformed EBWR-based recognisers. The short-burst "stop" class of phoneme models tended to have the worst recognition results.

For the Afrikaans recogniser adapted with English speech data, the fricative sounds were once again recognised the best by the MLLR and MAP transformed EBWR-based recognisers. The most commonly confused phoneme classes were found to be like-sounding diphthong classes.

Adapting the target-language models using MLLR followed by MAP techniques, and then employing embedded Baum-Welch re-estimation result in a considerable improvement in the cross-language phoneme recognition rate. Adapting the source-language recognisers with target language speech data brings the source-language speech models closer to the target language. This optimal use of target language data is the main reason for the improved cross-language phoneme recognition.

In the final set of experiments in Chapter 7, the effects of using pooled English and Afrikaans data and adapting these acoustic models is explored.

Chapter 7

DATA POOLING AND EBWR EXPERIMENTS

The final set of experiments follows the methodology suggested by Nieuwoudt [3, 8], whereby the mapped target language data is pooled with the original source language data. A new recogniser is then trained using this pool data. The models for the "bilingual" recogniser are then transformed using MAP adaptation to closer resemble the target language data. Finally, embedded Baum-Welch re-estimation (EBWR) is iteratively carried out on these transformed models.

This chapter is organised as follows:

- The first set of experiments in this Chapter involve pooling the mapped target language data with the original source language data to train a new recogniser.
- Next, the models for the "pooled" recogniser are then transformed using MAP adaptation to closer resemble the target language data. This is illustrated for the Afrikaans to English phoneme mapping case in Figure 7.1 below:

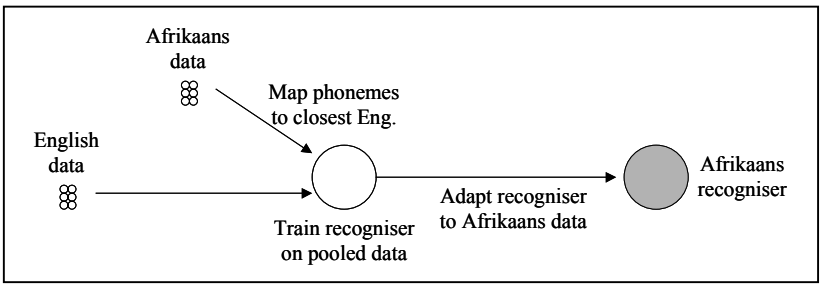

Figure 7.1: Illustration of data mapping, pooling and recogniser adaptation technique

- The final experiment investigates carrying out embedded Baum-Welch re-estimation (EBWR) on the transformed models. Once again the mapped data, provided by the nearest neighbour experiments that provided the best recognition results, is used for the re-estimation.

7.1. Effect of pooling English and Afrikaans mapped data followed by applying MAP adaptation

For this experiment, the SUN Speech Afrikaans training data is first mapped to the nearest TIMIT English phoneme. This mapped data is then pooled with the TIMIT English training data to form a training superset. This recogniser is then trained, increasing the mixture components from one to four. The new recogniser is then tested using the mapped SUN Speech test set, as well as the TIMIT SI and SX test set to assess it's "bilingual" nature. The models for this new recogniser are then transformed by MAP adaptation, using the mapped SUN Speech training set. The transformed recogniser is once again tested with the mapped SUN Speech test data, as well as the TIMIT SI and SX test set. The results appear in Table 7.1.

Table 7.1: Afrikaans to English phoneme mapping - Performance of the English recogniser on English data and mapped Afrikaans data per distance measure after MAP adaptation

Distance Measure		4 mix, no map, SUN Test	4 mix, no map, TIMIT test	After MAP SUN Test	After MAP TIMIT test
KL	%correct	47.78	60.82	50.34	54.56
	Accuracy	34.90	**55.26**	37.65	47.73
BHA	%correct	**47.95**	60.59	**51.63**	54.84
	Accuracy	**34.97**	55.14	**39.04**	48.02
MAH	%correct	32.90	59.01	36.47	48.75
	Accuracy	21.67	53.78	25.74	42.38
EUC	%correct	43.86	60.18	50.47	53.64
	Accuracy	31.69	54.96	37.09	47.49
L2	%correct	17.90	**60.84**	19.34	**58.90**
	Accuracy	13.40	55.10	14.83	**52.70**
JM	%correct	**47.95**	60.59	**51.63**	54.84
	Accuracy	**34.97**	55.14	**39.04**	48.02
Manual	%correct	42.24	60.00	46.12	51.26
	Accuracy	25.78	54.65	29.28	46.01

Table 7.1 makes for interesting reading. For the English recogniser trained on pooled data, the one using the mapped data derived by applying the Jeffreys-Matusita and Bhattacharyya distance measures outperform the others, with a recognition rate of 47.95% and an accuracy of 34.97% when tested on the mapped Afrikaans (SUN Speech) test data. The L2 distance metric retains the properties of the original TIMIT English phoneme set the best, and exhibits a phoneme recognition rate of 60.84% and an accuracy of 55.10% on the original TIMIT English phoneme set.

After MAP adaptation of the English recogniser has been carried out using the Afrikaans (SUN Speech) mapped training data, the adapted model based on the Jeffreys-Matusita and Bhattacharyya distance metrics outperform the other distance measures with a phoneme recognition rate of 51.63% and an accuracy of 39.04% on the mapped Afrikaans (SUN Speech) test set.

The pooled recogniser trained on the L2 mapped data is the best performing TIMIT phoneme

recogniser after MAP adaptation of the recogniser's models, with a TIMIT phoneme recognition rate of 58.90% and an accuracy of 52.70%.

7.2. Effect of pooling Afrikaans and English mapped data followed by applying MAP adaptation

In this experiment, the TIMIT English training data is mapped to the nearest SUN Speech phoneme. This mapped data is then pooled with the SUN Speech Afrikaans training data to form a training superset. This Afrikaans recogniser is then trained, increasing the mixture components from one to four. The new recogniser is then tested using the mapped English TIMIT SI and SX test set, as well as the Afrikaans SUN Speech test set to assess it's "bilingual" nature. The models for this new recogniser are then adapted by MAP adaptation, using a subset of the mapped TIMIT training set. The MAP-adapted recogniser is once again tested with the mapped TIMIT SI and SX test data, as well as the SUN Speech test set. The results appear in Table 7.2.

Table 7.2: English to Afrikaans phoneme mapping - Performance of the Afrikaans recogniser on Afrikaans data and mapped English data per distance measure after MAP adaptation

Distance Measure		4 mix, no map, TIMIT Test	4 mix, no map, SUN test	After MAP TIMIT Test	After MAP SUN test
KL	%correct	**55.83**	57.81	**59.61**	50.50
	Accuracy	**50.13**	45.04	**53.16**	35.10
BHA	%correct	55.53	58.01	59.30	50.35
	Accuracy	49.71	45.22	52.89	35.04
MAH	%correct	46.17	55.76	50.03	46.95
	Accuracy	40.74	42.56	44.36	32.13
EUC	%correct	53.70	57.77	58.05	52.00
	Accuracy	48.69	45.98	52.55	38.45
L2	%correct	36.67	**60.43**	36.87	**59.99**
	Accuracy	33.79	**46.86**	33.64	**45.95**
JM	%correct	55.53	58.01	59.30	50.35
	Accuracy	49.71	45.22	52.89	35.04

For the Afrikaans recogniser trained on pooled data, the one using the mapped data derived by applying the Kullback-Leibler distance measure outperforms the others, with a recognition rate of 55.83% and an accuracy of 50.13% when tested with the mapped English (TIMIT SI and SX) test data. In terms of performance, it is followed by the recognisers trained on the mapped Jeffreys-Matusita and Bhattacharyya distance metrics (recognition rate of 55.53% and accuracy of 49.71%). The L2 distance metric retains the properties of the original SUN Speech Afrikaans phoneme set the best, and exhibits a phoneme recognition rate of 60.43% and an accuracy of 46.86% on the original SUN Speech Afrikaans phoneme set.

After MAP adaptation has been carried out using the TIMIT mapped training data, the adapted model based on the Kullback-Leibler distance metric performs best with a phoneme recognition rate of 59.61%, and an accuracy of 53.16% on the mapped TIMIT SI and SX test set.

The pooled recogniser trained on the L2 mapped data is the best performing SUN Speech recogniser, with a SUN Speech phoneme recognition rate of 59.99% and an accuracy of 45.95%.

7.3. Effect of applying EBWR to model trained on Afrikaans mapped data and transformed by MAP

After the models have been adapted using MAP, embedded Baum-Welch re-estimation (EBWR) is iteratively applied three times. The performance of the re-estimated model is then tested on both the mapped Afrikaans (SUN Speech) test set, as well as on the original English (TIMIT SI and SX) test data.

Table 7.3: Afrikaans to English phoneme mapping - Performance of the English (pooled) recogniser on English data and mapped Afrikaans data per distance measure after MAP adaptation and embedded Baum-Welch re-estimation

Distance Measure		EBWR1 SUN Test	EBWR1 TIMIT Test	EBWR2 SUN Test	EBWR2 TIMIT Test	EBWR3 SUN Test	EBWR3 TIMIT Test
KL	%correct	54.61	47.32	56.01	46.17	56.45	45.62
	Accuracy	42.40	39.28	43.63	37.59	43.76	36.83
BHA	%correct	54.96	48.24	56.13	47.13	**56.80**	46.52
	Accuracy	43.05	40.19	43.75	38.47	**44.36**	37.75
MAH	%correct	44.05	38.76	45.95	37.57	46.82	37.21
	Accuracy	33.62	30.99	35.35	29.30	36.18	28.76
EUC	%correct	54.94	45.85	55.91	44.19	56.50	43.45
	Accuracy	40.27	36.87	41.05	34.12	41.23	33.10
L2	%correct	20.93	54.32	20.96	54.19	20.95	**54.15**
	Accuracy	17.23	47.45	17.15	47.19	17.19	**47.09**
JM	%correct	54.96	48.24	56.13	47.13	**56.80**	46.52
	Accuracy	43.05	40.19	43.75	38.47	**44.36**	37.75
Manual	%correct	50.00	37.86	51.44	34.90	52.36	34.22
	Accuracy	34.17	32.23	35.71	29.06	36.63	28.11

From Table 7.3, it is evident that the English (pooled) recognisers, trained on SUN Speech training data based on the Bhattacharyya and Jeffreys-Matusita distance measures, adapted by MAP adaptation and then re-estimated using embedded Baum-Welch re-estimation outperform all other distance measures with a final phoneme recognition rate after 3 EBWR iterations of 56.80% and an accuracy of 44.36% when tested on the mapped SUN Speech test data. This can now be put into the proper context and compared with the results obtained when the Afrikaans (SUN Speech) recogniser is trained on SUN Speech data and tested with SUN Speech data. This had a phoneme recognition rate of 63.71%. The 6.91% degradation in performance should be put into context of the amount of effort and time that has been spared in getting the recogniser up to this level. The data pooling process is undoubtedly just as time-consuming as training a recogniser from scratch. However, the difference lies in the robustness of the models generated. It would be expected that this recogniser trained on pooled data and then adapted and re-estimated would still maintain some of its salient recognition properties for the English TIMIT phoneme set.

As far as recognition rate on the original TIMIT SI and SX test set goes, the recogniser trained on the L2 metric pooled data outperformed the others, with a final phoneme recognition rate on the TIMIT test data of 54.15% and an accuracy of 47.09%.

Tables 7.4 and 7.5 list the phoneme classes with the best and worst recognition percentages per distance measure when using 39 MFCCs and after data pooling, MAP adaptation and 3 iterations of embedded Baum-Welch re-estimation for the SUN Speech to TIMIT phoneme mapping case.

Table 7.4: Top 5 recognition performances listed as a percentage of correctly recognised phonemes per distance measure for the Afrikaans (SUN Speech) to English (TIMIT) mapping after MAP adaptation and 3 iterations of EBWR

Distance Measure	**KL**	**BHA**	**MAH**	**EUC**	**L2***	**JM**	**Manual**
phoneme	uw	sh	dh	s	cl	sh	ch
%correct	92.3	86.8	83.8	91.7	22.5	86.8	96.1
phoneme	sh	f	ae	sh		f	oy
%correct	91.7	84.9	81.1	91.7		84.9	93.8
phoneme	ae	ae	aw	ay		ae	y
%correct	86.6	84.4	80.3	88.2		84.4	88.1
phoneme	f	cl	ah	n		cl	ay
%correct	84.2	82.2	76.1	87.4		82.2	85.3
phoneme	hh	hh	v	ch		hh	iy
%correct	82.3	82	75.7	85.7		82	82.5

* For the L2 metric, the [cl] phoneme was the closest phoneme for all SUN Speech phonemes

Table 7.5: Bottom 5 recognition performances listed as a percentage of correctly recognised phonemes per distance measure for the Afrikaans (SUN Speech) to English (TIMIT) mapping after MAP adaptation and 3 iterations of EBWR

Distance Measure	KL	BHA	MAH	EUC	L2*	JM	Manual
phoneme	m	m	oy	m	cl	m	dx
%correct	32.3	34.3	15.5	21.2	22.5	34.3	27
phoneme	ng	g	f	uh		g	uw
%correct	49.5	45.3	21	39.8		45.3	27.1
phoneme	g	uh	sh	ow		uh	b
%correct	50.8	47.2	33.3	42.9		47.2	30.8
phoneme	d	dx	d	b		dx	ow
%correct	50.9	50.2	34.5	44.4		50.2	36.9
phoneme	dx	d	jh	uw		d	ah
%correct	53.3	51.2	36.1	46		51.2	40.4

* For the L2 metric, the [cl] phoneme was the closest phoneme for all SUN Speech phonemes

From Table 7.4, it can be seen that the [sh], [f], [ch] and [cl] models were recognised the best. As in the previous set of experiments where MLLR and MAP adaptation was followed by embedded Baum-Welch re-estimation, the fricative sounds once again displayed the best recognition results. Note, however, that the [cl] (silence) class that in the previous set of experiments exhibited excellent recognition results, it displays average results with the recogniser trained on pooled data. This could be due to the data pooling leading to a more generalised initial model. As in the MLLR-MAP experiment set described earlier, while four of the six distance-based recognisers exhibited excellent recognition results for the [s] phoneme, the manual-mapped procedure had average phoneme recognition results for [s].

Table 7.5 shows that the [m], [dx], [d] and [uh] phoneme models displayed the poorest recognition results. This displays results similar to the MLLR-MAP experiment in the previous chapter.

Table 7.6: Confusion matrix information for Afrikaans to English phoneme mapping

	Most commonly confused phoneme classes per distance Measure					
Phoneme class	**KL**	**BHA**	**MAH**	**EUC**	**L2***	**JM**
m	ih	ih	th, cl	n, cl		ih
dx	cl, th	cl, th		cl, ah		cl, th
d	v, th	v, th	n, v			v, th
uh	cl, ng	ng, cl		dx, cl		ng, cl

* For the L2 metric, the [cl] phoneme was the closest phoneme for all SUN Speech phonemes

From Table 7.6, it is once again evident that in a large number of instances, the phoneme class is confused with the [cl] or "silence" model. As before, this can once again be attributed to the acoustic dissimilarities between the TIMIT and SUN Speech databases. It can also be seen from Table 7.6 that the nasal [m] phoneme (as in the word "**m**om") is often misclassified as the vowel [ih] phoneme, as in the word "b**i**t". It is also evident that the liquid [dx] (as in "mu**dd**y") and the stop [d] (as in "**d**og") are often confused with the fricative [th] sound (as in "**th**in"). The fact that all three of these phonemes belong to different categories should not obscure from the notion that they are acoustically similar.

7.4. Effect of applying EBWR to model trained on English mapped data and adapted by MAP

After the models have been adapted using MAP, embedded Baum-Welch re-estimation (EBWR) is iteratively applied three times. The performance of the re-estimated model is then tested on both the mapped TIMIT test set, as well as on the original SUN Speech test data.

Table 7.7: English to Afrikaans phoneme mapping - Performance of Afrikaans (pooled) recogniser on Afrikaans data and mapped English data per distance measure after MAP adaptation and embedded Baum-Welch re-estimation

Distance Measure		EBWR1 TIMIT Test	EBWR1 SUN Test	EBWR2 TIMIT Test	EBWR2 SUN Test	EBWR3 TIMIT Test	EBWR3 SUN Test
KL	%correct	61.68	45.43	61.87	44.45	61.95	43.88
	Accuracy	54.81	28.23	55.20	26.88	55.30	26.37
BHA	%correct	61.56	45.19	61.82	44.45	**61.91**	43.80
	Accuracy	54.65	28.58	55.03	27.42	55.08	26.95
MAH	%correct	55.02	38.62	56.31	37.30	56.88	37.03
	Accuracy	49.10	22.70	50.08	20.90	50.57	20.65
EUC	%correct	60.82	48.15	61.35	47.21	61.46	46.84
	Accuracy	54.93	33.61	55.32	32.75	**55.41**	32.41
L2	%correct	37.86	59.37	38.05	59.12	38.41	**59.03**
	Accuracy	34.45	45.16	34.65	44.90	34.98	**44.81**
JM	%correct	61.56	45.19	61.82	44.45	**61.91**	43.80
	Accuracy	54.65	28.58	55.03	27.42	55.08	26.95

From Table 7.7, it is evident that the Afrikaans (pooled) recogniser, trained on SUN Speech training data based on the Bhattacharyya and Jeffreys-Matusita distance measures, adapted by MAP adaptation and then re-estimated using embedded Baum-Welch re-estimation outperforms all other distance measures with a final phoneme recognition rate after 3 EBWR iterations of 61.91% and an accuracy of 55.08% when tested on the mapped TIMIT SI and SX test data. It is imperative to bear in mind that the only a portion of the TIMIT training data was used for the pooling, adaptation and re-estimation process. This can now be compared with the results obtained when the TIMIT recogniser is trained and tested with TIMIT data, yielding a phoneme recognition rate of 61.89%. The adapted and re-estimated recogniser using pooled data exhibits a marginal 0.02% improvement in phoneme recognition.

As far as recognition rate on the original SUN Speech test set goes, the recogniser trained on the L2 metric pooled data outperformed the others, with a final phoneme recognition rate on the SUN Speech test data of 59.03% and an accuracy of 44.81%.

Tables 7.8 and 7.9 list the phoneme classes with the best and worst recognition percentages per distance measure for the TIMIT English phoneme to SUN Speech Afrikaans phoneme mapping case, after data pooling, MAP adaptation, and 3 iterations of embedded Baum-Welch re-estimation have been carried out.

Table 7.8: Top 5 recognition performances listed as a percentage of correctly recognised phonemes per distance measure for TIMIT to SUN Speech mapping after MAP adaptation and 3 iterations of EBWR

Distance Measure	**KL**	**BHA**	**MAH**	**EUC**	**L2**	**JM**
phoneme	sil	sil	sil	sil	p	sil
%correct	90.3	90.3	88.3	91.4	97	90.3
phoneme	s	s	axi	s	sil	s
%correct	89.1	89.1	81.2	90.2	80.8	89.1
phoneme	f	f	n	k	jh	f
%correct	84.9	84.6	80.9	87.4	80.1	84.6
phoneme	w	w	u	oeu		w
%correct	81	79.3	76.6	78.6		79.3
phoneme	n	k	j	sh		k
%correct	77.6	77	74.3	76.7		77

Table 7.9: Bottom 5 recognition performances listed as a percentage of correctly recognised phonemes per distance measure for TIMIT to SUN Speech mapping after MAP adaptation and 3 iterations of EBWR

Distance Measure	**KL**	**BHA**	**MAH**	**EUC**	**L2**	**JM**
phoneme	ts	ts	ae	oi	s	ts
%correct	25.1	24.7	43.2	21.4	56.4	24.7
phoneme	oey	oey	oey	g	b	oey
%correct	35.7	34.9	47.9	38.1	35.9	34.9
phoneme	q	q	ax	oey		q
%correct	43.9	40.7	48.4	38.3		40.7
phoneme	o	b	b	a		b
%correct	48.1	48.9	51.5	40.6		48.9
phoneme	b	oi	p	d		oi
%correct	48.7	51.3	53.9	47.1		51.3

As in the MLLR-MAP experiment set in the previous chapter, only the best three recognised phoneme classes and the worst two recognised phoneme classes are provided for the L2 based recogniser. This is because the mapped SUN Speech phonemes were mapped to only five TIMIT phoneme classes during the mapping exercise.

From Table 7.8, it is clear that the [sil] (silence), [s], [f] and [w] models were recognised the best. In general, the fricative sounds were recognised the best by the MAP and EBWR adapted recognisers.

It can be seen from Table 7.9 that the [ts], [q], [b] and [oey] displayed the poorest recognition results.

Table 7.10: Confusion matrix information for English to Afrikaans phoneme mapping

	Most commonly confused phoneme classes per distance Measure					
Phoneme class	**KL**	**BHA**	**MAH**	**EUC**	**L2**	**JM**
ts	t, f	/, f	f, t, sil			/, f
q	k, t	k, t	s, ao			k, t
b	t, f	t, f	ao			t, f
oey	ax, axi	ax, axi	axi, o,ax	j, [ ], axi		ax, axi

It can be seen from Table 7.10 that the affricate [ts] phoneme (as in "ca**ts**") is often confused with the fricative [f] sound (as in "**f**in"). Recall that an affricate sound is produced when a stop and fricative consonant are both shortened and combined, which would explain this source of confusion. It is also evident that the stop [q] phoneme (as in the word "b**a**t") is regularly confused with the other stops [k] (as in "**k**ite") and [t] (as in the word "**t**ea"). Likewise, the [b] phoneme class (as in the word "**b**ee") is misclassified as the stop [t] and the fricative [f]. As has been previously found in this study, the diphthong [oey] phoneme (as in the word "b**ai**t") is most often confused with the vowel [ax] (as in "deb**i**t") and diphthong [axi] (as in the word "f**a**te") phoneme classes. The confusion between phonemes of the same category is to be expected due to the diversity that exists in the articulation of speakers.

7.5. Summary of Chapter findings

This chapter described the experiments that were performed during the data pooling

investigation and their results.

In the first experiment in this chapter, the SUN Speech Afrikaans training data is first mapped to the nearest TIMIT English phoneme, this mapped data is then pooled with the TIMIT English training data to form a training superset. The models for this new recogniser are then transformed by MAP adaptation, using the mapped Afrikaans SUN Speech training set. It was found that the adapted recogniser based on the Jeffreys-Matusita and Bhattacharyya distance metrics outperform the other distance measures.

The above experiment is repeated, but this time the English TMIT training data is mapped to the nearest SUN Speech Afrikaans phoneme. The mapped data is pooled with the Afrikaans data and then used to train a new recogniser. This new recogniser is then transformed by MAP adaptation, using the mapped English TIMIT training set. It was found that the adapted recogniser based on the Kullback-Leibler, Jeffreys-Matusita and Bhattacharyya distance metrics outperform the other distance measures.

The recognisers trained on the pooled training superset data from both the English and Afrikaans speech databases displayed the best “bilingual” phoneme recognition results in the study. This was to be expected as these phoneme models should be more robust since they were trained from scratch with data from both languages.

The final set of experiments in this chapter looked at performing Embedded Baum-Welch Re-estimation on each of the English and Afrikaans recognisers after MAP adaptation. This technique is found to further improve the cross-language recognition performance of the recognisers. As found previously, the recognisers that were adapted with data that was mapped using the Bhattacharyya and Jeffreys-Matusita distance metrics performed the best. As expected, the bilingual recognition performance of the recognisers dropped with each successive EBWR iteration as the recognisers become adapted closer to the target language.

For the English recogniser adapted with Afrikaans speech data, the fricative sounds were again recognised the best by the MAP and EBWR adapted recognisers. The silence model [cl]

tended to have the worst recognition results. This can be attributed to the acoustic dissimilarities between the TIMIT and SUN Speech databases.

For the Afrikaans recogniser adapted with English speech data, the fricative sounds were once again recognised the best by the MAP and EBWR-adapted recognisers. The most commonly confused phoneme classes were found to be the affricate [ts] sound which was confused with the fricative [f] sound. An affricate is produced through a combination of a stop and fricative consonant, indicating why this is likely to be confused with the fricative sound. The confusion between phonemes of the same category is to be expected due to the diversity that exists in the articulation of speakers.

The data pooling technique used to build a new recogniser using the automatically mapped phonemes from the target language as well as the source language phonemes produces a new recogniser which demonstrates moderate bilingual phoneme recognition capabilities. Adapting the "bilingual" recogniser further using MAP and embedded Baum-Welch re-estimation techniques results in the best cross-language phoneme recognition results. This combination of adaptation techniques together with the data pooling strategy is uniquely applied in the field of cross-language recognition. This data pooling followed by adaptation technique requires a considerably more time consuming training process. The adapted recogniser displays only slightly poorer phoneme recognition than the recognisers trained and tested on the same language database.

Chapter 8

CONCLUSION

This investigation has shown that an automatic phoneme mapping procedure can be used to map phonemes from a new target language to a base language for which a trained recogniser already exists. The current research in the cross-language speech recognition field has focused on manual methods performed by a phonetic expert to map phonemes. The new automated strategies proposed in this study are applied to English-to-Afrikaans phoneme mapping, as well as Afrikaans-to-English phoneme mapping. This has been previously applied to these language instances, but utilising manual phoneme mapping methods.

These experiments have also demonstrated that the choice of acoustic distance measure for the mapping does influence the results obtained. Four out of the six distance measures (the Kullback-Leibler measure, the Bhattacharyya distance metric, the Euclidean measure and the Jeffreys-Matusita distance), compared favourably with the manually undertaken phoneme mapping of a phonetic expert. Generally, the Bhattacharyya and Jeffreys-Matusita distance measures perform the best when mapping the phonemes to a target language, be this target language either English or Afrikaans. However, this study has shown that choosing the Euclidean or Kullback-Leibler distance measures will also result in good recognition results.

The relatively simple Euclidean metric exhibits good results and has shown that it is able to map the acoustic space of phoneme models relatively well.

The L2 and Mahalanobis distance measures, although useful in other pattern recognition applications, do not appear to be suitable for determining the degree of acoustic similarity between two phoneme classes. They have generally performed poorly in comparison with the other distance metrics.

A suggestion for the difference in cross-language phoneme recognition performance is due to the linearity of each of the distance measures. The distance measures that generally performed better were more linear and produced more consistently comparable distances between the phoneme classes. The measures that performed poorly, generally exhibited either very small or very large distance results between the phonemes, with the result that there was a degree of uncertainty about the whether the closest phoneme chosen was in fact a true indication of its closeness.

This investigation has also shown that the addition of the delta and acceleration MFCCs does generally improve the performance of the recogniser. This was demonstrated for both the Afrikaans to English phoneme mapping case, and vice versa. The general rule in speech recognition applications should be to always include these temporal features wherever possible.

In addition, the results of the study have shown that increasing the number of mixture components in the model does tend to aid phoneme recognition but only up to a limit. Thereafter, there is a slight degradation in phoneme recognition performance. The optimum number of mixture components varies per distance measure, but generally using two mixture components produces good results.

This study has also demonstrated that increasing the number of HMM states does not improve phoneme recognition performance. Although using all three states in the distance calculation produces better recognition results than only using the middle state, neither of these approaches is able to match the phoneme recognition performance of the single-state HMMs.

When single-state HMMs are used, all of the pertinent information is concentrated into the sole state. When three states are used, a new technique that is able to capture and utilise all the necessary information from each of the three states needs to be found. Applying the techniques employed in this dissertation do not fully extract the relevant features from the multi-state HMMs.

This study has also demonstrated the viability of using the phoneme mapping technique to generate seed models, and that the use of MLLR, MAP adaptation (for both the HMM's mean and variances) and embedded Baum-Welch model re-estimation techniques can then be effectively used to build a recogniser in a new language. This process requires much less effort and is considerably less time-consuming than if one had to completely rebuild the recogniser. Although a combination of MLLR and MAP techniques have been used previously in speech adaptation studies, the combination of MLLR, MAP and EBWR in cross-language speech recognition is a unique contribution of this study. Once again, this principle has been demonstrated for both the English to Afrikaans and Afrikaans to English case as well. However, there is a performance degradation that comes with the benefits of less effort and time to adapt the recogniser. There is an 13.96% degradation in phoneme recognition when compared with the results obtained when the SUN Speech recogniser is trained on SUN Speech data and tested with SUN Speech data (compared to adapting the English recogniser with Afrikaans data). For the English to Afrikaans mapping case (when the Afrikaans recogniser is adapted and re-estimated with TIMIT English data) there is a 18.20% reduction in phoneme recognition.

Finally, this investigation has verified that the automated phoneme mapping technique can also be applied to map the target language speech data into a common format as the source language, and so allow pooling of the speech data. Once the recogniser has been trained on the pooled data, it displays moderate "bilingual" capabilities, an aspect which has not been previously considered in the current research in this field. The recognisers trained on the pooled training superset data from both the English and Afrikaans speech databases displayed the best "bilingual" phoneme recognition results in the study. This was to be expected as these phoneme models should be more robust since they were trained from scratch with data from both languages.

Adapting the recogniser's models using MAP adaptation, followed by application of the embedded Baum-Welch re-estimation technique help to adapt the recogniser for phoneme recognition in the target language. This combination of adaptation techniques together with the data pooling strategy is uniquely applied in the field of cross-language recognition. After each successive adaptation cycle, the recogniser understandably loses its "bilingual" recognition abilities, and tends towards recognition in the target language, whether English or Afrikaans. For the adapted and re-estimated Afrikaans recogniser trained on pooled data, there is a 6.91% reduction in phoneme recognition (when tested on the mapped Afrikaans SUN Speech set) when compared to recogniser trained and tested with SUN Speech data alone. For the reverse case, when the English recogniser is trained on pooled SUN Speech and mapped TIMIT data, after adaptation and re-estimation, there is a marginal improvement in recognition of 0.02%.

It should also be borne in mind that while the data pooling followed by model adaptation technique has been shown to produce the best phoneme recognition performance, this is a more time-consuming process and is not always practical. However, it does produce more robust acoustic models.

One possible reason that could have contributed to the inferior cross-language recognition performance between the databases is due to the differences in the quality and recording conditions of the TIMIT and SUN Speech databases. Although cepstral mean subtraction has been employed in this investigation, this cannot possibly compensate for all the differences between the two speech databases.

Another factor that has possibly contributed to the results is the sensitivity of the mapping experiments to the number of occurrences of each phoneme in a set. The less the number of occurrences of each phoneme, the less general (and representative) the model derived for that particular phoneme set. This leads to less robust estimated acoustic models, which in turn leads to poorer mapping to the nearest cross-language phoneme class. This is especially relevant when mapping from the SUN Speech Afrikaans phonemes to the TIMIT English

phoneme set. The amount of available data does affect the results. As has been noted in [3], there are significant differences between the two databases, namely:

- TIMIT views stops as potentially two separate speech segments, a closure and a release. For example, an intervocalic stop of [t] would be transcribed as 'TCL T'. However, this transcription is based on the actual realisation, meaning that the affricate [ts] would be transcribed as 'TCL S'. The SUN Speech database, on the other hand, segments all phases of the stop together.
- The SUN Speech corpus makes provision for front rounded vowels which is the phonetic approach. In TIMIT, the vowels are handled in a more "phonemic" manner.
- In TIMIT, all vocalic sounds are grouped together and no diphthongisation is allowed. The SUN Speech database, in comparison, has an extensive set of diphthongs and labels quantity as well.
- The TIMIT transcriptions indicate the beginning and end of speech segments, with primary and secondary stress as well. This is absent in SUN Speech

A further factor that has influenced the outcome of the results in this investigation has been the inconsistency of the different phone inventories for the two speech corpora used. Although an attempt has been made to address this by trying to abide by the IPA-based phone inventory, there are inconsistencies.

The approach followed here can be extended to map between phonemes where same-language speech databases do not follow a consistent phoneme-labelling schema. As is often the case, merely mapping the label to another does not necessarily mean that the sounds labelled by the transcriptions are acoustically similar.

This investigation has shown that the cross-language performance of the recognisers does indeed compare favourably, but somewhat inferior with their respective recognition performances on same-language data. In addition, this study has demonstrated a viable technique to rapidly generate initial acoustic models for a new language. Once the adaptation techniques discussed previously have been implemented, the recognition performance of the new language recogniser, although slightly inferior to its same-language trained counterpart, demonstrates comparable phoneme recognition with considerably less training effort.

8.1 Future research

The investigation methodology could be improved by the addition of a threshold condition that compares two phoneme models and maps the phoneme models to each other only if the distance between them is below a predefined threshold. If the distance between the phoneme models is found to be greater than the threshold, then that source phoneme is not mapped but added as an additional phoneme class. The data pooling strategy discussed previously can then be carried out as before.

A further possibility that could improve cross-language phoneme recognition is the use of context-dependent phonemes. This has proven effective in past speech recognition experiments.

A final possibility is to investigate using HMM-neural network hybrids where the neural network part of the hybrid could be used in acoustic modeling and classification while the HMM part would be responsible for modeling the temporal nature of speech.

Appendix A

THE HTK TOOLKIT

The Hidden Markov Model Toolkit (HTK) V3.0 [9] was used during this investigation. HTK is a toolkit for building continuous density HMM-based recognisers. It is primarily intended for building sub-word based continuous speech recognisers and can be used in a wide range of pattern classification problems. HTK is built on an extensible modular library that simplifies the development of user-written tools. The toolkit includes signal processing functions, HMM training and testing tools, language modelling support and scoring software.

HTK consists of a set of tools that perform the different tasks in an HMM-based recognition system. These tools are written in C and C++ and make use of a library of basic functions for handling HMMs. Different data modules are used to transfer data between the different tools. These modules can contain speech data (as a waveform or as sequence of observation vectors), speech labelling data, the parameters that define HMMs or recognition networks.

The main tools in HTK are:

HEAdapt: This utility is used to perform adaptation of a set of HMMs using either maximum likelihood linear regression (MLLR) or maximum a-posteriori (MAP).
HBuild: The main purpose of this utility is to allow the expansion of HTK multi-level lattices and the conversion of bi-gram language models.
HLEd: Is a simple utility for manipulating label files.

HLStats: This utility reads in an HMM list and a set of label files. It then computes various statistics that are intended to assist in analysing acoustic training data and generating simple language models for recognition.

HInit: Is used to provide initial estimates for the parameters of a single HMM using a set of observation sequences.

HRest: performs basic Baum-Welch re-estimation of the parameters of a single HMM using a set of observation sequences.

HERest: is used to perform a single re-estimation of the parameters of a set of HMMs using an embedded training version of the Baum-Welch algorithm.

HVite: This is a general purpose Viterbi recogniser with syntax constraints and beam search.

HResults: This function takes a set of label files and compares them to the reference transcription files; it is the main performance analysis tool.

Appendix B

THE TIMIT AND SUN SPEECH DATABASES

The experiments are carried out using the TIMIT English database [11] and the SUN Speech English-Afrikaans corpus [12]. Only the SI (phonetically-diverse) and SX (phonetically-compact) TIMIT sentence sets were used. The TIMIT database contains about 80% more speech data than the English part of the SUN Speech database. There are 39 different phonemes listed in the TIMIT database (including the silence model) and a total of 59 phonemes used in the labelling of the SUN Speech database.

B.1. TIMIT database

The TIMIT acoustic-phonetic continuous speech corpus contains a corpus of read speech from 630 speakers from eight major dialects of American English. The database includes time-aligned orthographic, phonetic and word transcriptions as well as speech waveform data for each utterance.

The standard TIMIT database [11] makes use of a set of 61 phonemes. This normally reduced to a 39-phoneme set as shown in Table B1.

Table B.1: Mapping between 61 TMIT phonemes and 39 phoneme classes normally used in speech recognition experiments

New Phoneme	Original Phonemes	**New Phoneme**	Original Phonemes	**New Phoneme**	Original Phonemes	**New Phoneme**	Original Phonemes
aa	aa ao	**ae**	ae	**ah**	ah ax ax-h	**aw**	aw
ay	ay	**b**	b	**ch**	ch	**d**	d
dx	dx	**dh**	dh	**eh**	eh	**er**	er axr
ey	ey	**f**	f	**g**	g	**hh**	hh hv
ih	ih ix	**iy**	iy	**jh**	jh	**k**	k
l	l el	**m**	n	**p**	n en nx	**ng**	ng eng
ow	ow	**oy**	oy	**p**	p	**r**	r
s	s	**sh**	sh zh	**t**	t	**th**	th
uh	uh	**uw**	uw ux	**v**	v	**w**	w
y	y	**z**	z	**cl**	bcl pcl dcl tcl gcl kcl epi pau h#		

The following table lists the TIMIT phonemes, each categorised in terms of speech sounds, and lists English word examples.

Table B.2: Categorised TIMIT phonemes with English word examples

Category	TIMIT code	English word example
Vowels	ah	b**u**t
	iy	b**ee**t
	uw	b**oo**t
	eh	b**e**t
	ao	b**ough**t
	ax	**a**bout
	ih	b**i**t
	ae	b**a**t
Diphthongs	ay	b**i**te
	oy	b**oy**
	ey	b**ai**t
	ow	b**oa**t

Category	TIMIT code	English word example
	aw	b**ou**t
Nasals	m	**m**om
	n	**n**oon
	ng	si**ng**
Fricatives	f	**f**in
	hh	**h**ay
	s	**s**ea
	v	**v**an
	z	**z**one
	th	**th**in
	dh	**th**en
	sh	**sh**e
	zh	a**z**ure
Affricates	ch	**ch**oke
	jh	**j**oke
Glides	y	**y**acht
	w	**w**ay
Liquids	r	**r**ay
	l	**l**ay
	dx	mu**dd**y
Stops	b	**b**ee
	d	**d**ay
	g	**g**ay
	k	**k**ite
	p	**p**ea
	t	**t**ea
Other	cl	silence

B.2. SUN Speech database

The SUN Speech database [12] was compiled by the Department of Electrical and Electronic Engineering of the University of Stellenbosch. It contains phonetically labelled speech in both Afrikaans and English, although only the Afrikaans segment of the SUN Speech corpus was

used during this investigation. The speech data was recorded under controlled circumstances with 12bit resolution and a sampling rate of 16 kHz. Details of the number of speakers and the number of sentences spoken by each group of speakers are given in Table B.3. The 60 sentences comprising the four sentence sets were chosen to exhibit the diversity of phonemes in the two languages.

A total of 59 phonetic categories, including both a silence and unknown category, were used to segment both the Afrikaans and the English speech. However, for the Afrikaans segment of the database, there are only 54 phonetic categories.

Table B.3: Language and speaker composition for the SUN Speech database

Language	Speaker composition			Sentence Numbers
	Male	Female	Total	
English	55	21	76	21-60
Afrikaans	41	29	70	1-20

The following table lists the SUN Speech phonemes, each categorised in terms of speech sounds, and lists Afrikaans word examples.

Table B.4: Categorised SUN Speech phonemes with Afrikaans word examples

Category	SUN Speech Code	SUN Speech Code	Afrikaans word example
Vowels	97	a	k**a**t
	101	e	l**ee**s
	105	i	t**ie**r
	111	o	**oo**p
	117	u	s**oe**k
	121	y	n**uu**t
	130	eh:	s**ê**
	131	eh	m**e**t
	132	ao	k**o**s
	133	ao:	m**ô**re

Category	SUN Speech Code	SUN Speech Code	Afrikaans word example
	142	iax	kl**eu**r
	143	ax	**i**s
	144	ax:	w**î**e
	145	ae	**e**k
	149	oe	n**u**t
	150	oe:	br**û**e
	247	aa	**aa**n
Diphthongs	126	a:i	s**aai**
	128	o:i	m**ooi**
	140	ehi	b**e**djie
	151	axi	**y**s
	153	ui	m**oei**te
	210	iu:	l**eeu**
	211	oeu	**ou**d
	217	oey	l**ui**
Nasals	109	m	**m**at
	110	n	**n**et
	205	ng	s**ing**
Fricatives	102	f	**v**ars
	104	h	**h**uis
	115	s	**s**lim
	118	v	**w**as
	120	x	**g**aan
	122	z	**s**oem
	188	sh	**Sj**ina
	195	zh	**g**enre
Affricates	181	ts	
	191	ch	
	193	jh	
Liquids	114	r	
	82	R	**r**ooi
	94	[]	be**r**ge
	108	l	**l**ou
	218	/	refers to a flap
Glides	106	j	**j**as

Category	SUN Speech Code	SUN Speech Code	Afrikaans word example
	119	w	k**w**es
Stops	98	b	**b**ed
	100	d	**d**am
	103	g	ber**g**e
	107	k	**k**ar
	112	p	**p**os
	116	t	**t**aal
Other	42	sil	silence
	63	?	unknown

Appendix C

AFRIKAANS (SUN SPEECH) TO ENGLISH (TIMIT) PHONEME MAPPING

Table C.1: SUN Speech to TIMIT phoneme mapping per distance measure for 39 MFCCs, single-state HMMs

SUN Speech Code	ASCII Code	Distance Measure for TIMIT Mapping						
		KL	BHA	MAH	EUC	L2	JM	Manual
a	97	ah	ah	l	aw	p	ah	ah
aa	247	aa	aa	r	aa	t	aa	aa
ae	145	ae	ae	aw	ae	dh	ae	ae
ao	132	ah	l	l	oy	th	l	aa
a:i	126	ay	ay	ae	ay	p	ay	ay
ax	143	ih	ih	ah	dx	dh	ih	ah
axi	151	eh	eh	ay	eh	t	eh	ey
ax:	144	ae	uw	r	dx	eh	uw	er
b	98	v	v	d	v	p	v	b
ch	191	th	th	f	ch	dx	th	ch
d	100	d	d	z	cl	th	d	d
e	101	ey	ey	jh	ih	th	ey	ey
eh	131	ey	ey	l	ey	p	ey	eh
ehi	140	ey	ey	l	iy	aw	ey	eh

SUN Speech Code	ASCII Code	Distance Measure for TIMIT Mapping						
		KL	BHA	MAH	EUC	L2	JM	Manual
eh:	130	ey	uw	r	iy	aa	uw	eh
f	102	f	f	th	f	p	f	f
g	103	g	g	jh	v	cl	g	g
h	104	ow	ow	m	b	cl	ow	hh
i	105	ih	ih	eh	iy	cl	ih	iy
iax	142	ey	ey	r	ih	cl	ey	ey
iu:	210	uw	uw	r	uw	cl	uw	ow
j	106	ih	ih	aa	ih	cl	ih	y
jh	193	ch	jh	v	d	cl	jh	jh
k	107	th	th	dh	k	th	th	k
l	108	uh	uh	m	uh	cl	uh	l
m	109	ng	ng	r	m	cl	ng	m
n	110	uh	ng	r	n	cl	ng	n
ng	205	uh	uh	ch	n	cl	uh	ng
o	111	oy	oy	er	uh	cl	oy	ow
oe	149	ah	ah	ah	uh	cl	ah	ih
oeu	211	ay	ay	l	ah	ch	ay	ow
oey	217	eh	eh	aa	eh	cl	eh	ey
oi	134	ih	ih	jh	uh	cl	ih	oy
oqi	128	ow	ow	r	uh	cl	ow	oy
p	112	th	th	v	cl	cl	th	p
q	63	th	th	f	cl	cl	th	cl
R	82	dx	dx	oy	dx	cl	dx	r
[ ]	94	uh	uh	m	ah	cl	uh	r
r	114	dx	dx	ih	dx	s	dx	r
s	115	z	z	th	s	cl	z	sh
sh	188	sh	sh	th	sh	cl	sh	sh
sil	42	cl	cl	cl	cl	cl	cl	cl
/	218	dx	dx	z	v	cl	dx	dx
t	116	th	th	v	th	cl	th	t
ts	181	th	th	m	th	cl	th	t
u	117	m	m	eh	m	cl	m	uw
ui	153	iy	iy	sh	uw	cl	iy	ey
unk		th	th	f	cl	cl	th	cl

SUN Speech Code	ASCII Code	Distance Measure for TIMIT Mapping						
		KL	BHA	MAH	EUC	L2	JM	Manual
v	118	v	v	n	v	cl	v	v
w	119	l	l	r	ow	cl	l	w
x	120	hh	hh	m	p	cl	hh	hh
y	121	y	uw	r	iy	cl	uw	uw
z	122	z	z	hh	z	cl	z	z
zh	195	jh	jh	g	jh	cl	jh	sh
						cl		
*sil		hh	f	m	cl	cl	f	

Table C.2: SUN Speech to TIMIT phoneme mapping per distance measure for 26 MFCCs, single-state HMMs

SUN Speech Code	ASCII Code	Distance measure for TIMIT mapping						
		KL	BHA	MAH	EUC	L2	JM	Manual
a	97	aa	aa	ae	aa	cl	aa	ah
aa	247	d	dx	r	d	dh	dx	aa
ae	145	ae	eh	eh	ae	cl	eh	ae
ao	132	aw	aw	eh	p	cl	aw	aa
a:i	126	uw	uw	aa	uh	cl	uw	ay
ax	143	ng	uh	ow	n	cl	uh	ah
axi	151	jh	jh	jh	jh	cl	jh	ey
ax:	144	ch	cl	s	jh	cl	cl	er
b	98	d	th	f	s	th	th	b
ch	191	ch	sh	z	ch	th	sh	ch
d	100	th	th	f	th	th	th	d
e	101	ih	ih	aa	iy	th	ih	ey
eh	131	ey	eh	ae	eh	th	eh	eh
ehi	140	ih	ah	oy	uh	cl	ah	eh
eh:	130	eh	ae	eh	ae	cl	ae	eh
f	102	eh	uw	r	dh	cl	uw	f
g	103	ih	ih	eh	oy	cl	ih	g
h	104	ey	uw	aa	iy	cl	uw	hh
i	105	ey	ey	aa	ey	cl	ey	iy

SUN Speech Code	ASCII Code	Distance measure for TIMIT mapping						
		KL	BHA	MAH	EUC	L2	JM	Manual
iax	142	ih	ih	ah	oy	th	ih	ey
iu:	210	ow	l	eh	ow	cl	l	ow
j	106	ey	ey	aa	ih	cl	ey	y
jh	193	ey	iy	eh	y	th	iy	jh
k	107	ow	ih	ah	ow	th	ih	k
l	108	ay	ay	aw	aa	cl	ay	l
m	109	z	z	m	s	th	z	m
n	110	iy	iy	ah	iy	cl	iy	n
ng	205	f	k	n	cl	th	k	ng
o	111	v	ih	ah	cl	cl	ih	ow
oe	149	dh	g	d	k	th	g	ih
oeu	211	ow	uh	eh	m	cl	uh	ow
oey	217	th	th	th	cl	th	th	ey
oi	134	z	z	n	z	cl	z	oy
oqi	128	dx	ih	r	iy	cl	ih	oy
p	112	th	th	th	cl	th	th	p
q	63	uh	uh	oy	ah	cl	uh	cl
R	82	ih	uh	ah	n	cl	uh	r
[ ]	94	m	uh	uh	m	cl	uh	r
r	114	ah	uh	r	ah	cl	uh	r
s	115	v	th	th	cl	th	th	sh
sh	188	ih	uh	oy	uh	dh	uh	sh
sil	42	cl	cl	cl	cl	cl	cl	cl
/	218	ow	ih	r	k	cl	ih	dx
t	116	ng	g	d	v	th	g	t
ts	181	f	f	th	f	th	f	t
u	117	oy	iy	oy	oy	cl	iy	uw
ui	153	hh	dh	dx	dx	th	dh	ey
unk		k	v	n	b	th	v	cl
v	118	ae	ay	ae	ae	cl	ay	v
w	119	oy	dx	r	uh	cl	dx	w
x	120	dh	w	w	uh	th	w	hh
y	121	hh	ih	ch	k	th	ih	uw
z	122	f	f	f	cl	cl	f	z

SUN Speech Code	ASCII Code	Distance measure for TIMIT mapping						
		KL	BHA	MAH	EUC	L2	JM	Manual
zh	195	oy	th	th	ah	cl	th	sh
*sil	42	ey	iy	ae	iy	cl	iy	

Table C.3: SUN Speech to TIMIT phoneme mapping per distance measure for 13 MFCCs, using single-state HMMs

SUN Speech Code	ASCII Code	Distance measure for TIMIT mapping						
		KL	BHA	MAH	EUC	L2	JM	Manual
a	97	aa	aa	ay	aa	dh	aa	ah
aa	247	ih	ih	ah	v	dh	ih	aa
ae	145	oy	oy	b	oy	p	oy	ae
ao	132	ow	ow	ay	ah	dh	ow	aa
a:i	126	ey	ey	aw	oy	dh	ey	ay
ax	143	ng	hh	ow	ng	cl	hh	ah
axi	151	jh	jh	ch	jh	th	jh	ey
ax:	144	ch	ch	ch	jh	th	ch	er
b	98	jh	jh	ch	jh	cl	jh	b
ch	191	ch	ch	ch	ch	th	ch	ch
d	100	th	th	t	th	cl	th	d
e	101	ey	iy	ay	iy	th	iy	ey
eh	131	oy	oy	ih	oy	dh	oy	eh
ehi	140	ah	ah	ay	v	ah	ah	eh
eh:	130	ay	ay	ay	ah	cl	ay	eh
f	102	ah	ih	eh	v	cl	ih	f
g	103	ah	ah	eh	ah	v	ah	g
h	104	uh	uh	aw	oy	hh	uh	hh
i	105	ey	ey	aa	iy	cl	ey	iy
iax	142	ow	ow	eh	b	th	ow	ey
iu:	210	l	l	ay	l	th	l	ow
j	106	ey	ey	ay	ey	th	ey	y
jh	193	ey	hh	ay	ng	th	hh	jh
k	107	ow	ow	aw	b	th	ow	k

SUN Speech Code	ASCII Code	Distance measure for TIMIT mapping						
		KL	BHA	MAH	EUC	L2	JM	Manual
l	108	aw	aw	ay	ay	cl	aw	l
m	109	z	z	s	z	dh	z	m
n	110	ey	iy	ay	iy	th	iy	n
ng	205	p	p	k	cl	cl	p	ng
o	111	cl	cl	dh	cl	f	cl	ow
oe	149	dh	dh	m	dh	dh	dh	ih
oeu	211	ow	uw	ay	uw	v	uw	ow
oey	217	t	t	t	cl	th	t	ey
oi	134	s	s	m	s	th	s	oy
oqi	128	d	d	ow	iy	dh	d	oy
p	112	dh	dh	dh	cl	cl	dh	p
q	63	uh	uh	ay	ih	dh	uh	cl
R	82	ng	uw	ow	n	dh	uw	r
[ ]	94	m	m	er	m	v	m	r
r	114	v	v	uw	v	cl	v	r
s	115	cl	cl	cl	cl	th	cl	sh
sh	188	d	d	ah	dx	dh	d	sh
sil	42	cl	cl	cl	cl	cl	cl	cl
/	218	hh	hh	ow	hh	v	hh	dx
t	116	g	g	b	v	d	g	t
ts	181	f	f	th	cl	cl	f	t
u	117	oy	oy	eh	oy	cl	oy	uw
ui	153	dx	dx	oy	dx	cl	dx	ey
unk		cl	cl	cl	cl	th	cl	cl
v	118	ay	ay	ay	ay	cl	ay	v
w	119	d	d	er	d	dh	d	w
x	120	ow	ow	r	b	th	ow	hh
y	121	t	t	t	cl	th	t	uw
z	122	cl	cl	cl	cl	th	cl	z
zh	195	hh	hh	hh	hh	th	hh	sh
*sil	42	iy	iy	ay	iy	dh	iy	

Appendix D

ENGLISH (TIMIT) TO AFRIKAANS (SUN SPEECH) PHONEME MAPPING

Table D.1: TIMIT to SUN Speech phoneme mapping per distance measure for 39 MFCCs, using single-state HMMs

TIMIT code	Distance measure for SUN Speech mapping					
	KL	BHA	MAH	EUC	L2	JM
aa	aa	aa	o	aa	b	aa
ae	axi	axi	oey	axi	b	axi
ah	ax	ax	ax	[ ]	b	ax
aw	aa	aa	o	a	b	aa
ay	a	a	oey	a:i	b	a
b	g	g	ax	oi	b	g
ch	ch	ch	ao	sh	b	ch
cl	sil	sil	sil	sil	sil	sil
d	t	t	ax	/	b	t
dh	R	R	p	R	b	R
dx	t	t	t	/	b	t
eh	oey	oey	i	oey	b	oey
er	ax	ax	u	oeu	b	ax
ey	eh	eh	oey	eh	b	eh
f	f	f	ts	t	b	f
g	g	g	w	g	b	g

TIMIT code	Distance measure for SUN Speech mapping					
	KL	BHA	MAH	EUC	L2	JM
hh	q	q	b	d	b	q
ih	ax	ax	i	j	b	ax
iy	i	i	axi	y	b	i
jh	zh	zh	ao	zh	b	zh
k	k	k	ao	k	b	k
l	ao	ao	aa	ao	b	ao
m	n	m	q	m	b	m
n	n	n	s	n	b	n
ng	ng	ng	q	n	b	ng
ow	o	ao	oey	ao	b	ao
oy	oi	oi	ae	ao	b	oi
p	k	k	ao	k	p	k
r	o	o	ao	oeu	jh	o
s	s	s	n	s	b	s
sh	sh	sh	j	sh	b	sh
t	t	t	ao	t	b	t
th	ts	ts	ts	t	b	ts
uh	ax	ax	ax	oi	b	ax
uw	u	u	ao	iu:	b	u
v	b	b	p	g	s	b
w	w	w	w	ao	b	w
y	y	y	j	y	b	y
z	s	s	m	s	b	s
*cl	p	p	p	p	b	p

Table D.2: TIMIT to SUN Speech phoneme mapping per distance measure for 26 MFCCs, using single-state HMMs

TIMIT code	Distance measure for SUN Speech mapping					
	KL	BHA	MAH	EUC	L2	JM
aa	a	a	ao	a	ui	a
ae	eh	eh	v	eh	ui	eh
ah	q	g	ax	r	ui	g
aw	v	v	g	v	ui	v
ay	l	l	iu:	l	ui	l

TIMIT code	Distance measure for SUN Speech mapping					
	KL	BHA	MAH	EUC	L2	JM
b	t	t	UNK	iax	ax:	t
ch	ch	ch	r	ch	ui	ch
cl	sil	sil	sil	sil	sil	sil
d	s	s	d	o	ui	s
dh	aa	aa	axi	aa	ui	aa
dx	UNK	UNK	UNK	oe	ui	UNK
eh	ae	ae	R	eh	ui	ae
er	aa	aa	q	ao	s	aa
ey	j	j	u	j	ui	j
f	ts	ts	b	oey	s	ts
g	t	t	ui	t	ui	t
hh	ui	ui	ui	y	s	ui
ih	g	g	r	o:i	ui	g
iy	sil	sil	q	n	ui	sil
jh	axi	axi	oi	axi	ui	axi
k	s	s	UNK	s	ui	s
l	iu:	iu:	u	iu:	s	iu:
m	[ ]	[ ]	y	[ ]	ui	[ ]
n	R	R	b	R	ui	R
ng	ax	ax	oi	R	ui	ax
ow	iu:	iu:	u	iu:	ui	iu:
oy	iax	iax	n	iax	ui	iax
p	s	s	ax:	s	ui	s
r	aa	aa	sh	ao	ui	aa
s	oi	oi	oi	oi	s	oi
sh	ch	ch	zh	ch	s	ch
t	oey	oey	b	oey	ui	oey
th	d	d	p	oey	s	d
uh	q	q	ax	iax	ui	q
uw	oeu	oeu	q	a:i	ui	oeu
v	UNK	UNK	p	t	ui	UNK
w	x	x	x	x	s	x
y	n	n	x	n	ui	n
z	oi	oi	zh	oi	s	oi

TIMIT code	Distance measure for SUN Speech mapping					
	KL	BHA	MAH	EUC	L2	JM
*cl	p	p	ui	p	p	p

Table D.3: TIMIT to SUN Speech phoneme mapping per distance measure for 13 MFCCs, using single-state HMMs

TIMIT code	Distance measure for SUN Speech mapping					
	KL	BHA	MAH	EUC	L2	JM
aa	a	a	a:i	a	b	a
ae	eh	eh	ae	eh	iu:	eh
ah	v	v	w	r	p	v
aw	l	l	u	v	o:i	l
ay	v	v	ao	l	o:i	v
b	s	s	ax:	iax	p	s
ch	axi	axi	s	ch	p	axi
cl	sil	sil	sil	sil	sil	sil
d	s	s	ch	o	oey	s
dh	aa	aa	t	aa	[ ]	aa
dx	o	o	ax:	o	b	o
eh	eh	eh	eh	eh	o:i	eh
er	aa	ao	o	ao	p	ao
ey	h	h	/	j	d	h
f	ts	ts	y	oey	b	ts
g	s	s	axi	t	p	s
hh	s	s	ch	y	o:i	s
ih	iax	iax	i	o:i	[ ]	iax
iy	sil	sil	UNK	n	iu:	sil
jh	axi	axi	s	axi	ui	axi
k	s	s	ch	s	iu:	s
l	iu:	iu:	iax	iu:	b	iu:
m	t	t	s	[ ]	o:i	t
n	UNK	UNK	ch	R	o:i	UNK
ng	t	t	s	R	o:i	t
ow	iu:	iu:	/	iu:	b	iu:
oy	iu:	iu:	iu:	iax	b	iu:
p	ng	ng	ch	s	d	ng

TIMIT code	Distance measure for SUN Speech mapping					
	KL	BHA	MAH	EUC	L2	JM
r	v	x	w	ao	p	x
s	oi	oi	s	oi	iu:	oi
sh	axi	axi	ax:	ch	p	axi
t	d	d	oi	oey	p	d
th	d	d	zh	oey	p	d
uh	iax	iax	UNK	iax	ui	iax
uw	iax	iax	UNK	a:i	b	iax
v	zh	zh	oey	t	p	zh
w	iu:	iu:	t	x	b	iu:
y	i	i	s	n	ui	i
z	oi	oi	s	oi	p	oi
*cl	s	s	b	p	p	s

BIBLIOGRAPHY

[1] L. Couvreur and J. Boite "Speaker tracking in broadcast audio material in the framework of the THISL Project," *Proceedings of 1999 Workshop on Accessing Information in Spoken Audio (ESCA-ETRW)*, Cambridge, UK, pp.84-89, April 1999.

[2] B. Mak and E. Barnard, "Phone clustering using the Bhattacharyya distance," *Proceedings of the 4th International Conference on Spoken Language Processing*, Vol. 4, Philadelphia, USA, pp. 2005-2008, Oct. 1996.

[3] C. Nieuwoudt, "Cross-language acoustic adaptation for automatic speech recognition," *PhD Thesis*, University of Pretoria, South Africa, April 2000.

[4] M. Falkhausen, H. Reininger and D. Wolf "Calculation of distance measures between hidden Markov models," *Proceedings of the European Conference on Speech Technology and Communication*, Madrid, Spain, pp. 1487-1490, 1995.

[5] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[6] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. COM-15, pp. 52–60, 1967.

[7] J. Kohler "Comparing three methods to create multilingual phone models for vocabulary independent speech recognition tasks," *ESCA-NATO Workshop on Multilingual Interoperability in Speech Technology*, Leusden, The Netherlands, pp. 79-84, 1999.

[8] C. Nieuwoudt and E. Botha, "Cross-language use of acoustic information for automatic speech recognition," *Speech Communication*, Vol. 38, pp. 101-113, Sept. 2002.

[9] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK Book, " *HTK Version 3.0*, Microsoft Corporation, July 2000.

[10] S. Pissarra, C. Ribeiro, L.V. Dutra, C.D. Renno and J.V. Soares, "Culture classification using polarimetric information from SIR-C/X-SAR mission: Bebedouro region, Brazil," *Technical Report SIR-C/X-SAR*, 1996.

[11] ARPA, "The DARPA TIMIT acoustic-phonetic continuous speech corpus," *NIST Speech Disc CD1-1.1*, Dec. 1990.

[12] Department of Electrical and Electronic Engineering - University of Stellenbosch, South Africa, "*The SUN Speech Database*," 1997.

[13] J.J. Sooful and E.C. Botha, "An acoustic distance measure for automatic cross-language phoneme mapping*," Proceedings of the Twelfth Annual Symposium of the Pattern Recognition Association of South Africa*, Franschhoek, South Africa, pp. 99-102, 29-30 November 2001.

[14] J.J. Sooful and E.C. Botha, "Comparison of acoustic distance measures for automatic cross-language phoneme mapping," *Proceedings of the 10th International Conference on Spoken Language Processing*, Denver, Colorado, pp. 521-524,September 2002.

[15] J.W.F. Thirion "Phoneme recognition with HTK on the TIMIT database," *African Speech Technology Technical Report*, University of Pretoria, South Africa, 23 Feb. 2001.

[16] J.W.F. Thirion "HTK installation instructions under Windows NT," *African Speech Technology Technical Report*, University of Pretoria, South Africa, 23 Feb. 2001.

[17] F. de Wet, "Isolated word speech recognition in Xhosa*," MScEng Thesis*, University of Pretoria, South Africa, February 1999.

[18] R. Schalkoff, *Pattern Recognition: Statistical, structural and neural approaches*, John Wiley & Sons, USA, 1992.

[19] R. Dugad and U.B. Desai, "A tutorial on hidden Markov models," *Technical Report: Signal processing and artificial neural networks laboratory – Indian Institute of Technology*, Mumbai, India, May 1996.

[20] C. J. Leggetter and P. C. Woodland, "Speaker adaptation using linear regression," *Technical Report CUED -TR.181*, University of Cambridge Eng. Department, Cambridge, UK, June 1994.

[21] Y. Grenier, L. Miclet, J. C. Maurin, and H. Michel, "Speaker Adaptation for Phoneme Recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 3, pp. 1273-1275, Atlanta, USA, 1981.

[22] F. Kubala, R. Schwartz, and C. Barry, "Speaker Adaptation from a Speaker Independent Training Corpus," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 137-140, Albuquerque, USA, 1990.

[23] D.A. Reynolds, T.F. Quatieri and R.B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing 10*, pp. 19–41, USA, 2000.

[24] J.-L. Gauvain and C.-H. Lee, "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains," *IEEE Transactions On Speech and Audio Processing*, Vol.2, pp. 291-298, 1994.

[25] J.-L. Gauvain and C.-H. Lee, "MAP Estimation of Continuous Density HMM: Theory and Applications," *Proceedings of the DARPA Speech and Natural Language Processing Workshop*, pp. 185-190, Harriman, New York, Feb. 1992.

[26] J.P. Campbell Jr., "Speaker recognition: A tutorial," *Proceedings of the IEEE – Special Issue on Automated Biometrics*, Vol. 85, No. 9, New Jersey, USA, pp. 1436-1462, Sept. 1997.

[27] R. Huber and H.A. Mayer, "ERC – Evolutionary Resample and Combine for Adaptive Parallel Training Data Set Selection," *Proceedings of International Conference on Pattern Recognition*, Brisbane, Australia, Aug. 1998.

[28] P.H. Swain and R.C. King, "Two effective feature selection criteria for multispectral remote sensing," *The International Joint Conference on Pattern Recognition*, Washington, D.C., Nov. 1973.

[29] R. Willerman and P.K. Kuhl, "Cross-language speech perception: Swedish, English, and Spanish speakers' perception of front rounded vowels," *Proceedings of the 4th International Conference on Spoken Language Processing,* New York, USA, pp. 442-445, Oct. 1996.

[30] P. Dalsgaard, O. Andersen, H. Hesselager and B. Petek, "Language-identification using Language-dependent Phonemes and Language-independent Speech Units," *Proceedings of the 4th International Conference on Spoken Language Processing*, New York, USA, pp. 1808-1811 Oct. 1996.

[31] A. Acero and X. Huang, "Speaker and Gender Normalization for Continuous-Density Hidden Markov Models," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, USA, Volume 1, pp. 342-345, May 1996.

[32] S. Goronzy and R. Kompe, "A MAP-like Weighting Scheme for MLLR Speaker Adaptation," *Proceedings of the European Conference on Speech Technology and Communication*, Budapest, Hungary, pp. 5-8, 1999.

[33] C. Leggetter and P. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of CD HMMs," *Computer Speech and Language*, Volume 9, pp. 171-185, 1995.

[34] O. Andersen, and P. Dalsgaard, "Language Identification based on Cross-language Acoustic Models and Optimised Information Combination," *Proceedings of the European Conference on Speech Technology and Communication*, Rhodes, Greece, pp. 67-70, 1997.

[35] B. Wheatley, K. Kondo, W. Anderson, and Y. Muthusamy, "An Evaluation of Cross-language Adaptation For Rapid HMM Development in a new language" *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Adelaide, Australia, pp. 237-240, 1994.

[36] T. Schultz and A. Waibel, "Multilingual and Crosslingual Speech Recognition," *Proceedings of the DARPA Broadcast News Transcription and Understanding*, Lansdowne, Virginia, pp 259-262, Feb. 1998.

[37] A. Waibel, H. Soltau, T. Schultz, T. Schaaf and F. Metze, "Multilingual Speech Recognition," Verbmobil: Foundations of Speech-to-Speech Translation, Springer Verlag, 2000.

[38] A. Waibel, P. Geutner, L. Mayfield-Tomokiyo, T. Schultz, and M. Woszczyna, "Multilinguality in Speech and Spoken Language Systems," *Proceedings of the IEEE, Special Issue on Spoken Language Processing*, Volume 88, No.8, pp 1297-1313, August 2000.

[39] T. Schultz and A. Waibel, "Fast bootstrapping of LVCSR systems with multilingual phoneme sets," *Proceedings of the European Conference on Speech Technology and Communication*, Rhodes, Greece, pp. 371-374, 1997.

[40] P. Fung, C. Y. Ma and W. K. Liu, "MAP-based cross-language adaptation augmented by linguistic knowledge: from English to Chinese," *Proceedings of the European Conference on Speech Technology and Communication*, Budapest, Hungary, pp. 871-874, 1999.

[41] A. Lindstrom and R. Eklund, "How foreign are "foreign" speech sounds? Implications for speech recognition and speech synthesis," *Multi-Lingual Interoperability in Speech Technology, RTO Meeting Proceedings 28*, Québec, Canada, pp. 15–19, Aug. 2000.

[42] G. Stemmer, E. Nöth and H. Niemann, "Acoustic Modeling of Foreign Words in a German Speech Recognition System," *Proceedings of the European Conference on Speech Technology and Communication*, Aalborg, Denmark, pp. 2745-2748, Sept. 2001.

[43] B. McMurray, M. Spivey, and R. Aslin, "The perception of consonants by adults and infants: Categorical or categorized?," *University of Rochester Working Papers in the Language Sciences*, Volume 1, No.2, pp. 215-256, 2000.

[44] M. Macon, A. Cronk and J. Wouters, "Generalization and discrimination in tree-structured unit selection," *Proceedings of the Third International Workshop on Speech Synthesis*, Blue Mountains, Australia, pp. 195-200, 1998.

[45] E.A. Marta, and L.V. Sá, "Auditory Cells with Frequency Resolution Sharper than Critical Bands Play a Role in Stop Consonant Perception: Evidence from Cross-Language Recognition Experiments," *Proceedings of NATO ASI on Computational Hearing*, Il Ciocco, Italy, pp. 173-179, July 1998.

[46] C.-H. Jo, T. Kawahara, S. Doshita, and M. Dantsuji, "Automatic Pronunciation Error Detection And Guidance For Foreign Language Learning," *Proceedings of the International Conference on Spoken Language Processing*, Sydney, Australia, Volume 6, pp. 2639-2942, December 1998.

[47] H. Cerf-Danon, S. De Gennaro, M. Feretti, J. Gonzalez and E. Keppel, "TANGORA - A Large Vocabulary Speech Recognition System for Five Languages," *Proceedings of the European Conference on Speech Communication and Technology*, Genova, Italy, Volume 1, pages 183-186, Sept. 1991

[48] N. Mukherjee, N. Rajput, L. V. Subramaniam and A. Verma, "On Deriving a Phoneme Model for a New Language," *Proceedings of the IEEE International Conference on Spoken Language Processing*, Beijing, China, Oct. 2000.

[49] S. Rapp, "Automatic Phonemic Transcription and Linguistic Annotation from Known Text with Hidden Markov Models / An Aligner for German," *Proceedings of ELSNET goes East and IMACS Workshop*, Moscow, pp. 152-163, 1995.

[50] H. G. Hirsch and D. Pearce, "The AURORA Experimental Framework for the Performance Evaluations of Speech Recognition Systems under Noisy Conditions,"

Proceedings of the International Conference on Spoken Language Processing 2000, Beijing, China, Vol. 4, pp 29-32, 2000.

[51] C.C. Wooters and A. Stolcke, "Multiple-pronunciation Lexical Modeling in a Speaker-independent Speech Understanding System," *Proceedings of International Conference on Spoken Language Processing*, Yokohama, Vol. 3, pp. 1363-1366, 1994.

[52] C. Vogler and D. Metaxas, "Adapting Hidden Markov Models for ASL Recognition by Using Three-dimensional Computer Vision Methods," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Orlando, USA, pp. 156-161, October 1997.

[53] M. Ostendorf, V. Digalakis and O. Kimball, "From HMMs to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, Vol. 4, No. 5, pp. 360-378, September 1996.

[54] S. Bengio and J. Mariethoz, "Comparison of Client Model Adaptation Schemes," *Technical Report IDIAP-RR 01-25*, IDIAP, 2001.

[55] I. Potamitis, N. Fakotakis and G. Kokkinakis, "Independent Component Analysis Applied to Feature Extraction for Robust Automatic Speech Recognition," *Electronics Letters*, Vol. 36, No.23, pp. 1977-1979, Nov 2000.

[56] S.-J. Doh, "Enhancements to Transformation-Based Speaker Adaptation: Principal Component and Inter-Class Maximum Likelihood Linear Regression," *Ph.D Thesis*, ECE Department, CMU, July 2000.

[57] U. Uebler, "Speech recognition in 7 languages," *Proceedings of the Workshop on Multi-Lingual Interoperability in Speech Technology*, Leusden, The Netherlands, 1999.

[58] J.E. Hamaker, "MLLR: A speaker adaptation technique for LVCSR," Institute for Signal and Information Processing, Lecture Notes - Department of Electrical and Computer Engineering, November 1999.

[59] H. Ye, P. Fung and T. Huang, "Principal Mixture Speaker Adaptation for Improved Continuous Speech Recognition," *Proceedings of the International Conference on Spoken Language Processing 2000*, Beijing, China, 2000.

[60] J. Tebelskis, "Speech Recognition using Neural Networks," *PhD Thesis*, Carnegie Mellon University, USA, May 1995.

[61] G. Rigoll, "Hybrid Speech Recognition Systems: A Real Alternative To Traditional Approaches?" In *Survey Lecture, Proceedings of International Workshop Speech and Computer (SPECOM'98),* St. Petersburg, Russia, pp. 33-42, October 1998.

[62] F.T. Johansen, "A Comparison of Hybrid HMM Architectures using Global Discriminative Training," *Proceedings of International Conference on Spoken Language Processing*, Philadelphia, pp. 498-501, October 1996.

[63] K. Torkkola, "Improved speech recognition using learning vector quantization," In E. Fiesler and R. Beale, editors, *The Handbook of Neural Computation*, Chapter G1.6. Oxford University Press and Institute of Physics Publishing, 1997.