

Fast Accurate Diphone-Based Phoneme Recognition

MARIANNE DU PREEZ



*Thesis presented in partial fulfilment of the requirements for the degree Master of
Science in Electronic Engineering at the University of Stellenbosch*

SUPERVISORS: Prof. J. A. du Preez
and Dr. H. A. Engelbrecht
March 2009

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work unless indicated otherwise, and that I have not previously in its entirety or in part submitted it at any university for a degree.

SIGNATURE

DATE

Abstract

Statistical speech recognition systems typically utilise a set of statistical models of subword units based on the set of phonemes in a target language. However, in continuous speech it is important to consider co-articulation effects and the interactions between neighbouring sounds, as over-generalisation of the phonetic models can negatively affect system accuracy. Traditionally co-articulation in continuous speech is handled by incorporating contextual information into the subword model by means of context-dependent models, which exponentially increase the number of subword models. In contrast, transitional models aim to handle co-articulation by modelling the interphone dynamics found in the transitions between phonemes.

This research aimed to perform an objective analysis of diphones as subword units for use in hidden Markov model-based continuous-speech recognition systems, with special emphasis on a direct comparison to a context-dependent biphone-based system in terms of complexity, accuracy and computational efficiency in similar parametric conditions. To simulate practical conditions, the experiments were designed to evaluate these systems in a low resource environment – limited supply of training data, computing power and system memory – while still attempting fast, accurate phoneme recognition.

Adaptation techniques designed to exploit characteristics inherent in diphones, as well as techniques used for effective parameter estimation and state-level tying were used to reduce resource requirements while simultaneously increasing parameter reliability. These techniques include diphthong splitting, utilisation of a basic diphone grammar, diphone set completion, maximum a posteriori estimation and decision-tree based state clustering algorithms. The experiments were designed to evaluate the contribution of each adaptation technique individually and subsequently compare the optimised diphone-based recognition system to a biphone-based recognition system that received similar treatment.

Results showed that diphone-based recognition systems perform better than both traditional phoneme-based systems and context-dependent biphone-based systems when evaluated in similar parametric conditions. Therefore, diphones are effective subword units, which carry suprasegmental knowledge of speech signals and provide an excellent compromise between detailed co-articulation modelling and acceptable system performance.

Opsomming

Statistiese spraakherkenning maak tipies gebruik van 'n stel statistiese subwoordmodelle wat gebaseer is op die stel foneme in 'n gegewe taal. Dit is egter belangrik in kontinue spraak om ko-artikulasie en interaksie van naburige klanke in ag te neem, aangesien 'n oorveralgemening van fonetiese modelle die stelselakkuraatheid negatief kan beïnvloed. Tradisioneel word hierdie ko-artikulasie effekte hanteer deur middel van konteks-afhanklike modellering waar foneme in verskillende kontekste apart gemodelleer word. Hierdie proses veroorsaak 'n eksponensiële groei in die aantal subwoord modelle. In teenstelling hiermee kan oorgangsmodelle gebruik word om die ko-artikulasie te hanteer deur die dinamika in die oorgange tussen foneme vas te vang.

Hierdie navorsing het beoog om 'n objektiewe analise van difone as subwoordmodelle in verskuilde Markov model-gebaseerde kontinue-spraakherkenningstelsels te doen. Spesiale klem is geplaas op 'n direkte vergelyking van die difoonstelsel met 'n konteks-afhanklike bifoonstelsel in terme van kompleksiteit, akkuraatheid en effektiewe verwerkingsvermoe in parametries eenderse toestande. Om praktiese toestande te simuleer is alle eksperimente ontwerp om die stelsels te evalueer in 'n omgewing met min hulpbronne – 'n beperkte hoeveelheid afrigdata, verwerkingskrag en stelselgeheue – maar steeds te mik vir vinnige, akkurate foneemherkenning.

Aanpassingstegnieke is ontwerp en gebruik om difoonmodelle te optimeer deur hulpbronvereistes te verminder en terselfdetyd parameter-betroubaarheid te verhoog. Hierdie tegnieke sluit in die aanwending van difooneienskappe deur middel van diftong-verdeling, gebruik van 'n basiese difoongrammatika, difoonstel-voltooiing, *maximum a posteriori* beraming en toestandsgroepering deur middel van beslissingsbome. Die eksperimente is ontwerp om die bydrae van elke tegniek individueel te ontleed, waarna die beste stelsel vergelyk word met 'n bifoonstelsel wat op 'n soortgelyke manier hanteer is.

Resultate dui daarop dat difoon-gebaseerde stelsels beter vaar as beide tradisionele foneem- en konteks-afhanklike bifoon-gebaseerde stelsels in soortgelyke parametriese toestande. Difone is dus effektiewe subwoord-eenhede wat supra-segmentele inligting bevat en is 'n uitstekende kompromis tussen gedetailleerde ko-artikulasie modellering en aanvaarbare stelsel-werkverrigting.

Acknowledgements

I would like to thank my study leaders, Prof. du Preez and Dr. Engelbrecht, for their extensive help, guidance and support without which this thesis would have been impossible. Their expertise, suggestions and feedback were of immeasurable value.

I want to express my gratitude to my family and friends for their encouragement and moral support.

Special thanks goes to my mother for using her extensive wisdom and knowledge to help with the grammatical editing and proofreading of this document. She has provided assistance in numerous ways and helped shape and improve the end result.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Background	2
1.2.1	Statistical Speech Recognition	2
1.2.2	Elementary Linguistic Theory	3
1.2.3	Acoustic Modelling for use in Speech Recognition	8
1.3	Literature Synopsis	10
1.4	Objectives	12
1.5	Contributions	13
1.6	Thesis Overview	13
1.6.1	Background Theory on Statistical Speech Recognition	13
1.6.2	Analyses of Diphones and their Use in Speech Recognition	14
1.6.3	Experiments, Results and Conclusions	15
2	Speech Recognition: Theoretical Background	18
2.1	Types of Speech Recognition	18
2.2	Literature Study	21
2.2.1	A Brief History	21
2.2.2	The Use of Diphones in Speech Recognition	23
2.3	The Speech Recognition System	29
2.3.1	Mathematical Formulation	29
2.3.2	Components of the Speech Recognition System	30
2.3.3	Digital Signal Processing for Speech Signals	32
2.3.4	Acoustic Modelling	37
2.3.5	Lexical Modelling	39
2.3.6	Language Modelling	39
2.4	Summary	40

3	Hidden Markov Model Theory	42
3.1	Definition of a Hidden Markov Model	43
3.1.1	Markov Chains	43
3.1.2	Hidden Markov Models	44
3.2	Algorithms Used with Hidden Markov Models	46
3.2.1	The Evaluation Problem	47
3.2.2	The Decoding Problem	50
3.2.3	The Learning Problem	52
3.3	Hidden Markov Models Used in Speech Recognition	54
3.3.1	HMM Topology	54
3.3.2	State Output Probability Distributions	57
3.4	Implementation of Acoustic Modelling for Phoneme Recognition	61
3.4.1	Creating and Training the Model	61
3.4.2	Alignment of the Labelled Training Set	63
3.4.3	Decoding	64
3.4.4	Evaluation	66
3.5	Summary	67
4	Diphones as Base Units for Speech Recognition	68
4.1	Speech Units Used in Linguistics	68
4.1.1	Syllables	69
4.1.2	Monophones	71
4.1.3	Biphones	72
4.1.4	Triphones	73
4.1.5	Diphones	73
4.2	Modelling Transitions versus Modelling Context Dependency	74
4.2.1	Trainability	75
4.2.2	Complexity and Resource Requirements	78
4.2.3	Handling Inter-word Contexts	79
4.2.4	Modelling of Unseen Contexts	79
4.3	Implementation Strategies for Diphones	80
4.3.1	Non-parametric Methods	80
4.3.2	Parametric Methods	82
4.3.3	Automatic Diphone Segmentation	84
4.4	Acoustic Modelling with Diphones as Base Unit	84
4.4.1	Segmentation	85
4.4.2	Model Structure	85

4.4.3	Decoding	86
4.4.4	Evaluation	86
4.5	Summary	87
5	Adaptation Techniques for Diphone Models	89
5.1	Diphthong Splitting	89
5.2	Basic Diphone Grammar for Phoneme Spotting	90
5.3	Diphone Set Completion	93
5.3.1	Building Diphone Models from Well-trained Monophone Models . .	93
5.3.2	Bootstrapping the Diphone Set with Monophone models	94
5.4	Maximum A Posteriori Estimation	94
5.4.1	Mathematical Formulation	95
5.4.2	MAP Estimation of Gaussian Mean Values	96
5.4.3	MAP Estimation as Used in this Thesis	97
5.5	Decision Tree Based State Clustering	97
5.5.1	Overview of Decision Tree Logic	98
5.5.2	Classification and Regression Trees (CART)	100
5.5.3	Creating a CART	101
5.5.4	Classification and Regression Trees as Used in this Thesis	103
5.6	Summary	104
6	Experimental Investigation	106
6.1	Experimental Setup	106
6.1.1	Hardware Platform	106
6.1.2	Software Platform	106
6.1.3	The AST Data set	107
6.1.4	Signal Processing	107
6.1.5	Statistical Modelling Parameters	108
6.1.6	System Evaluation	109
6.1.7	Statistical Significance Tests	109
6.2	Monophone-based Continuous Phoneme Recognition	110
6.2.1	Motivation	110
6.2.2	Experimental Setup	111
6.2.3	Results	111
6.2.4	Interpretation	112
6.3	Diphone-based Continuous Phoneme Recognition	112
6.3.1	First Approximations	114

6.3.2	Diphthong Splitting	117
6.3.3	MAP Estimation	119
6.3.4	Decision-tree Based State Clustering	124
6.3.5	Interpretation of Diphone Results	128
6.4	Biphone-based Continuous Phoneme Recognition	130
6.4.1	Motivation	130
6.4.2	Experimental Setup	131
6.4.3	Results	134
6.4.4	Interpretation	134
6.5	Comparison of Systems in Limited-Resource Environments	135
6.6	Summary	136
7	Conclusions	138
7.1	Concluding Perspective	138
7.2	Context Within Existing Research	139
7.3	Future Work	140
7.3.1	Diphone-based System	140
7.3.2	Comparison with a Triphone-based System	141
A	Selected Topics from Linguistic Theory	152
A.1	International Phonetic Alphabet	152
A.2	Types of Phonemes	154
A.3	Additional Terms Related to the Production of Speech Sounds	156
B	Speech Corpus	158
B.1	The African Speech Technology (AST) Speech Corpus	158
B.1.1	Data sets	158
B.1.2	Collection Parameters	158
B.1.3	Phoneme Set	159
B.2	Subword Unit Statistics	159
B.2.1	Monophones	159
C	CART	162
C.1	Question Set	162
C.2	Pruning	164
C.3	Minimum Description Length Based Induction and Pruning	165

List of Figures

1.1	Time waveform of the word “test”	6
1.2	Enlargement of the steady-state region of the vowel /E/	6
1.3	Enlargement of the steady-state region of the consonant /s/	6
1.4	General discrete-time filter model for speech production	8
2.1	System diagram of the basic components in a speech recognition system . .	31
2.2	Relationship between the Mel frequency scale and the Hertz frequency scale	36
3.1	3 State left-to-right Hidden Markov Model	46
3.2	The forward algorithm. The partial probability $\alpha_{t+1}(j)$ is recursively defined by multiplying the output probability of state s_j with the sum of the partial probabilities of the states leading to state s_j multiplied by their respective transition probabilities.	49
3.3	3 State fully connected Hidden Markov Model	55
3.4	5 State left-to-right Hidden Markov Model with non-emitting starting and terminating states	56
3.5	A two-dimensional Gaussian distribution	58
3.6	Parallel-HMM model used for the decoding of phoneme sequences in continuous phoneme recognition	65
4.1	Acoustic waveform, spectrogram and corresponding subdivision into different subword units for the utterance “Lend me your ears”. The subword units from top to bottom are phonemes, left-context biphones, right-context biphones and diphones.	70
5.1	Basic diphone grammar for the language defined by monophones “SIL”, “a”, “b” and “c”. The spotter assumes the existence of silence at the beginning and end of each utterance. The black null states are called <i>cluster states</i> and represent the set of diphones with a common first monophone. .	92

5.2	Construction of a diphone model from two well-estimated left-to-right monophone models.	94
5.3	A partitioned two-dimensional input space that has been divided into five regions using axis-aligned boundaries and the corresponding binary decision tree.	100
6.1	Summary of diphone-based experiments designed to isolate contributions by various adaptation techniques. Coloured boxes represent unique configurations of the diphone-based system for which experiments were run. Each branch represents a system design decision made, each of which can be attributed to an adaptation technique as described in Chapter 5	113
6.2	Diphones built from monophone models and stitched back together in the diphone spotter. Subfigure a) shows a sequence of three phonemes and the states that will be used to build two diphone models corresponding to the transitions between the phonemes. Subfigure b) shows the constructed diphone models next to each other, clearly showing the duplicated state b_2 . When these two diphone models are linked in the diphone spotter structure, the configuration shown in subfigure c) occurs where the new “monophone b” is not equivalent to the original “monophone b”.	121
6.3	Inserting skiplinks between two diphone models in the spotter structure to remove the duplicate state. The original monophone model is shown in a). The two halves of the diphone models built from the original monophone are connected in such a way as to create an HMM model equivalent to the original monophone model by adding skiplinks between them as shown in b). The null state coloured black would have been the original null state connecting the two diphone models.	122
6.4	Accuracies and execution times of model sets A, B and C after decision tree-based state clustering at different state occupancy intervals. The optimal system is one yielding high recognition accuracy with the lowest possible number of parameters, resulting in faster decoding.	126
6.5	Accuracies and execution times of model sets A, B and C after decision tree-based state clustering as a function of the eventual density set size. . .	127

List of Tables

2.1	Summary of research into the use of diphone templates for speech recognition	24
2.2	Summary of research into the use of hybrid HMM/template systems for speech recognition	25
2.3	Summary of research into the use of diphone subspace models for speech recognition	26
2.4	Summary of research into the use HMM/ANN hybrid systems for speech recognition	27
4.1	Comparison of modelling characteristics on state-level between a left-context biphone, triphone and diphone model in similar parametric conditions. All models are assumed to be three-state, left-to-right hidden Markov models.	77
6.1	Duration statistics for the training and testing data sets from the English subset of the AST speech database.	107
6.2	Continuous Recognition Accuracy: Monophone Baseline System	112
6.3	Continuous Recognition Accuracy: First Approximation Diphone System .	116
6.4	Decoding Execution Time: First Approximation Diphone System (Execution Time is depicted as hh:mm:ss.ms) with additional value in terms of real-time (RT)	116
6.5	Continuous Recognition Accuracy: Diphone System after Diphthong Splitting	118
6.6	Decoding Execution Time: Diphone System after diphthong splitting (Execution Time is depicted as hh:mm:ss.ms) with additional value in terms of real-time (RT)	119
6.7	Continuous Recognition Accuracy: Diphone System Bootstrapped from monophones with Additional MAP Estimation	123

6.8	Decoding Execution Time: Diphone System Bootstrapped from monophones with Additional MAP Estimation (Execution Time is depicted as hh:mm:ss.ms) with additional value in terms of real-time (RT)	123
6.9	Continuous Recognition Accuracy: Best results for each of the three diphone-based systems used in the CART experiment	125
6.10	Decoding Execution Time of best results for each of the three diphone-based systems used in the CART experiment (Execution Time is depicted as hh:mm:ss.ms) with additional value in terms of real-time (RT)	128
6.11	Continuous Recognition Accuracy: Recognition systems based on left-context biphone models	134
A.1	Phoneme Chart : Vowel and Consonant Sounds for South African English .	152
B.1	Occurrence and Duration statistics for monophone labels found in the training and testing subsets of the English AST Data set	159
C.1	Set definitions used in CART question set	163

Chapter 1

Introduction

1.1 Motivation

The choice of speech unit used in continuous-speech recognition greatly influences the accuracy, complexity, robustness and expandability of a speech recognition system. Although most systems make use of phonetic subword units, research in the field of acoustic phonetics has shown that phonetic transitions are more important for the perception of speech than the phonetic units themselves [11]. The transitions contain more suprasegmental information and are better suited to model the fast-changing dynamic characteristic of human speech.

The modelling of phonetic units such as phonemes is often preferred because it is generally a small set, reducing the complexity of the speech recognition system and increasing computational efficiency. The implementation of grammatical rules – used to define valid combinations of phonetic units to create words and sentences in a specific language – are simple and easily adaptable. However, the high level of generality in this type of system results in lower recognition scores. The great variability of the same speech sounds uttered in different contexts presents a major challenge when creating parametric models for these sounds, leading to the preference of context-dependent modelling techniques to incorporate contextual information into the acoustic models. These context-dependent techniques include the use of biphones for either left- or right-context modelling and triphones for the simultaneous consideration of both the preceding and following phone contexts.

The explicit modelling of the transitions between phonetic units provides an alternative to context-modelling techniques. The interphone dynamics are modelled through the use of diphone units, which are used to model the transition from the centre of one phone to the centre of the next phone. Diphones are used extensively in speech synthe-

sis applications because of the improved fluidity of speech achieved when concatenating transitions between phones, rather than concatenating the phone segments themselves.

The purpose of this research was to examine diphones as basic subword units for use with continuous-speech recognition. Specifically, the diphones are subjected to adaptation techniques, which are used to increase system accuracy while still approximating the computational efficiency of phonetic subword units. The use of diphones in speech recognition systems is currently limited due to the relatively small gain in accuracy over phonetic systems when compared to context-dependent modelling. However, in most practical speech recognition systems training data is scarce and computational resources are low. In these conditions the diphone-based system may well outperform a more complex context-modelled system, while still providing better recognition scores than the basic phonetic system.

1.2 Background

This thesis is concerned with the acoustic modelling phase of a specific type of speech recognition known as *statistical speech recognition*. This section discusses elementary theory pertaining to statistical speech recognition, as well as linguistic theory and acoustic modelling concepts that are necessary to understand the methodologies in this research and place it in context with previous research.

1.2.1 Statistical Speech Recognition

The three major types of speech recognition techniques are the *acoustic-phonetic* approach, the *pattern recognition* approach and the *artificial intelligence* approach.

- The acoustic-phonetic approach assumes that each phonemic unit can be broadly characterised by a set of variables (e.g. pitch, voiced/unvoiced, formant frequencies) that can be used to segment and label speech signals through the use of pattern matching. This approach relies completely on knowledge of the physical production of speech and the transmission of the sound waves.
- The pattern recognition approach requires little explicit knowledge of speech production, using a large set of data to train generic parametric models. These parametric representations of patterns are then used for comparisons in order to classify them. Pattern recognition techniques include template matching and the use of *Hidden Markov Models* (HMMs) and *Artificial Neural Networks* (ANNs). Statistical speech

recognition is a type of pattern recognition technique utilising statistics to extract the information needed to train and represent parametric models.

- The artificial intelligence approach attempts to mechanise the speech recognition procedure to mimic the way the human brain perceives and processes speech signals.

1.2.2 Elementary Linguistic Theory

Speech recognition is by its very nature inter-disciplinary, requiring knowledge from linguistic theory to provide an effective framework for the implementation, which is based on techniques from mathematics, statistics and computer science. This section provides a brief overview of important terms and concepts in linguistic theory as referenced in this research. Chapter 4 contains a detailed discussion of linguistic theory applied to acoustic modelling. The implementation done for this research is handled in Chapter 5. Additional definitions of related terms and concepts can be found in Appendix A, along with a list of English phonemes in the *International Phonetic Alphabet* notation.

Theoretical linguistics

Linguistics is the scientific study of language [61]. It is a social science using aspects of biology, psychology, anthropology and sociology to understand the role language plays in human society. Theoretical linguistics refers to our understanding of linguistic knowledge, including speech production, structure and meaning, whereas applied linguistics refers to the application of linguistic theory to real-world problems such as language acquisition and conversation analysis. There are several different disciplines within the field of theoretical linguistics.

- Phonetics is the study of the production, transmission and perception of speech sounds.
- Phonology is the study of sound patterns in a language and how the sounds influence each other based on structure.
- Morphology is the study of word formation, structure and the rules governing it.
- Syntax is the study of sentence structure and the grammatical rules that apply in a language.
- Semantics is the study of meaning in speech and how we convey it.

- Prosody is the study of rhythm, stress and intonation of speech, revealing information pertaining to the intent of the sentence (for example a question or command) and the emotional state of the speaker.
- Pragmatics is the study of language in a social setting, indicating that we communicate with more than just the words we use.

Different areas of theoretical linguistics are used in different areas of the speech recognition system:

- Phonology is used in acoustic modelling to extract parameters related to the physical properties of speech sounds,
- Morphology is used in lexical modelling to assist in the construction of words from subword units and
- Syntax is used in language modelling to define the word-transition rules that govern a specific language.

Areas in applied linguistics such as language acquisition, sociolinguistics, psycholinguistics and cognitive linguistics are used in the broader contexts of machine learning and machine translation.

Acoustic Theory of Speech Production

A sound pressure wave is created by forcing air from the lungs through a series of structures constituting the human speech production system [24]. The main sections are the *trachea* (windpipe), *pharyngeal cavity* (throat), *oral cavity* (mouth) and the *nasal cavity* (nose), creating spaces with unique acoustic contributions. Finer anatomical components, called articulators, move to different positions and configurations to change the sound wave into different speech sounds. The articulators include the *vocal chords*, *soft palate*, *tongue*, *teeth* and *lips*. The speech production system can be seen as an acoustic filtering operation where an excitation signal is filtered by the cavities and articulators to change the signal properties. The excitation signal can be either *voiced* or *unvoiced*. Voiced sounds are produced by forcing air through the opening between the vocal folds, with the resulting vibration frequency related to the vocal fold tension. Unvoiced sounds are produced by constricting airflow from the lungs at some point in the vocal tract and producing turbulence.

The spectral characteristics of speech signals vary over time due to the continuous physical changes occurring during speech production. But, because of physical limita-

tions related to the speed of articulatory movement, short segments of sounds are quasi-stationary, possessing similar acoustic properties for short periods. These short segments are identified as either vowels (voiced sounds with no restriction of airflow) or consonants (voiced or unvoiced sounds with significant restriction of airflow).

To further our understanding of its characteristics, the speech signal is analysed in the time and frequency domains. The time waveform (example shown in Figure 1.1) can be used to determine the intensity, periodicity, duration and boundaries of individual speech sounds. The enlargements of the vowel /E/ in Figure 1.2 and the consonant /s/ in Figure 1.3 in the spoken word “test”, illustrate the differences between these sounds. The /E/ sound is sonorant, producing more energy and containing a periodic component produced by the vibrations of the vocal chords. The /s/ sound is a fricative consonant, produced by restricting airflow to produce a more random, noisy speech pattern. Figure 1.1 shows that continuous speech is not a string of individual well-formed sounds. It can rather be seen as a sequence of target sounds with the transitions between these targets forming the largest percentage of the speech signal. This is due to physical restraints on the movement speed of the articulators needed to produce the sounds. The transitional sounds are highly dependent on the preceding and following sounds, leading to various differences in utterances of the same sound in continuous speech. This effect is called *co-articulation* and is an important aspect to consider when creating acoustic models.

Phones and Phonemes

Two branches of theoretical linguistics is particularly important for acoustic modelling: phonetics and phonology.

Phonetics is concerned with the physical articulation of speech sounds, the acoustic properties of the sound waves and the perception of speech sounds by the human ear and brain without distinguishing between different languages. The smallest units that are distinguishable in phonetics are called *phones* – a collection of finite, mutually exclusive sounds, each with corresponding articulatory gestures.

Phonology is the study of the realisations of phones (called *phonemes*) in continuous speech – their context, interactions and meaning in a specific language. Phones that are acoustically slightly different from each other, but provide exactly the same function in a specific language are called allophones and are grouped together to form a phoneme, creating a set of unique sounds that distinguish meaning in a specific language. To illustrate allophones, consider the “t”-sound in the words *tip* and *stand*. In the first case the phone [t^h] is aspirated whereas in the second, the phone [t] is not aspirated. Although these phones sound slightly different, they do not distinguish meaning in the

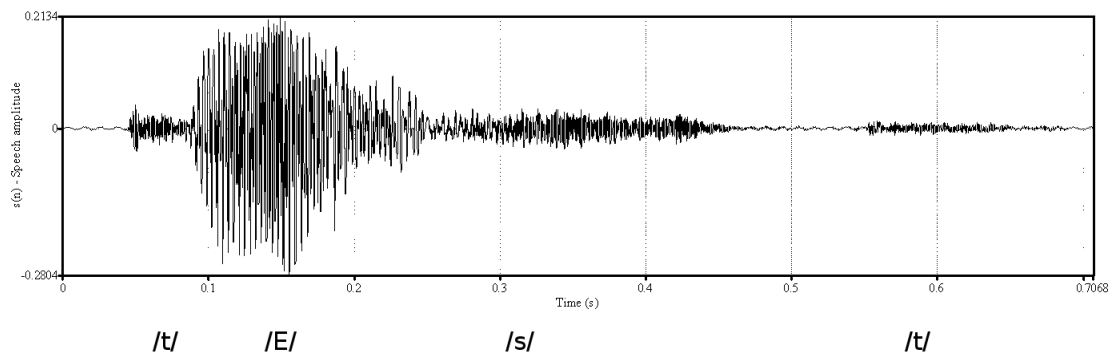


Figure 1.1: *Time waveform of the word “test”*

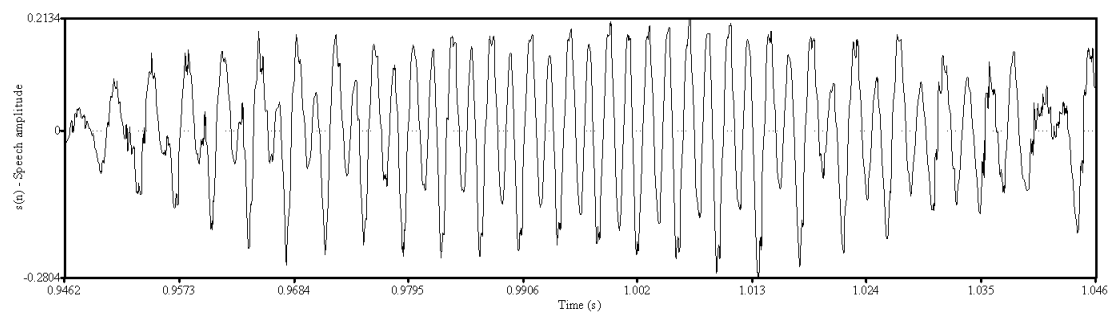


Figure 1.2: *Enlargement of the steady-state region of the vowel /E/*

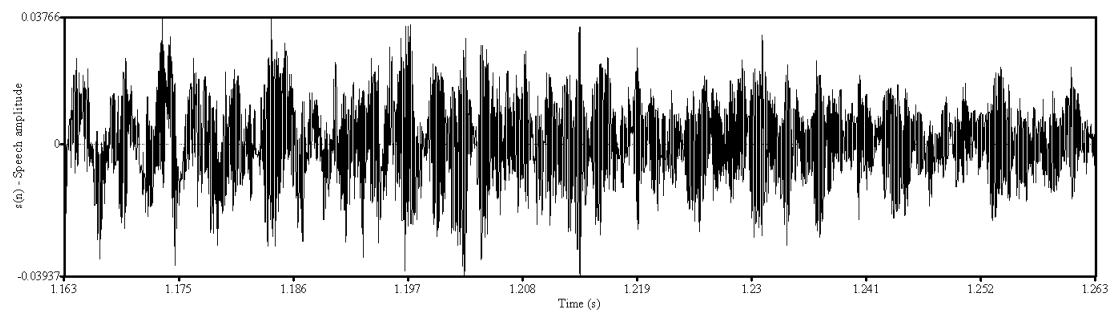


Figure 1.3: *Enlargement of the steady-state region of the consonant /s/*

English language and are therefore grouped together in the phoneme /t/. A simple test to determine whether two phones are allophones or not is to find a minimal pair - two valid words that differ only by the phones in question.

The International Phonetic Alphabet (IPA) provides a standard for the notation of phones and phonemes of all languages. According to the IPA notation, phones are usually enclosed in square brackets (e.g. [t]) whereas phonemes are enclosed in virgules (e.g. /t/). The English language contains around thirteen to twenty-two vowels, including diphthongs, and twenty-two to twenty-six consonants, creating a set of between 35 and 48 phonemes. These variations are due to the choice of phone-groupings with short and long versions of the same sounds either grouped together or kept apart. To accommodate words pronounced differently in an English dialect or foreign sounds not usually found in English, a different phoneme set may be more appropriate than one suitable for standard English.

A summary of English phonemes, their use in this research and word examples are given in Appendix A.

Co-articulation

Due to physical limitations on how fast the articulators can move and the relative speed and variability of continuous speech communication, phones typically overlap and influence each other rather than forming a discrete sequence of sounds. Co-articulation causes changes in phoneme articulation and acoustics. If the articulator configuration for the following phoneme does not conflict with the current phoneme, the articulators will start moving into position early in anticipation, changing the acoustic properties of the current phoneme. When moving on to the next phoneme, the articulators needed for the previous phoneme can now move from their previous positions to participate in the production of the current and future speech sounds. This continuous change of articulator positions leads to many variations of the same phoneme, depending on its context. The target articulator configuration of each phoneme can be found somewhere in the centre of the phoneme, with the beginning and end of the phoneme highly influenced by its neighbouring phonemes. The exact position and duration of the target configuration also depend on the phoneme context. To deal with co-articulation effects in continuous speech, the recognition system either has to make use of transitional models, such as diphones, or incorporate context-modelling by using biphone or triphone models. These two methodologies are discussed in Chapter 4.

1.2.3 Acoustic Modelling for use in Speech Recognition

Discrete-Time Modelling

Early research based on the resonant structure of cylindrical tubes showed the analogy between acoustical systems and electric transmission lines, leading to the description of the speech production process as a discrete-time transfer function. Figure 1.4 shows a general linear discrete-time model for speech production first described by Rabiner and Schafer in 1987 [68]. This model represents the speech production process based on characteristics of the output signal, while disregarding coupling or nonlinear effects between the subsystems in the model. In this filter model, the vocal-tract model $H(z)$ and radiation model $R(z)$ are excited by a discrete-time glottal excitation signal $u_{glottis}(n)$. During unvoiced speech activity, the excitation source is a flat spectrum noise source modelled by a random noise generator. During voiced speech activity, the excitation uses an estimate of the local pitch period to set an impulse train generator that drives a glottal pulse shaping filter $G(z)$ [24]. We can therefore assume that the output pressure wave of the speech production system is the result of filtering the appropriate excitation by a sequence of linear, separable filters.

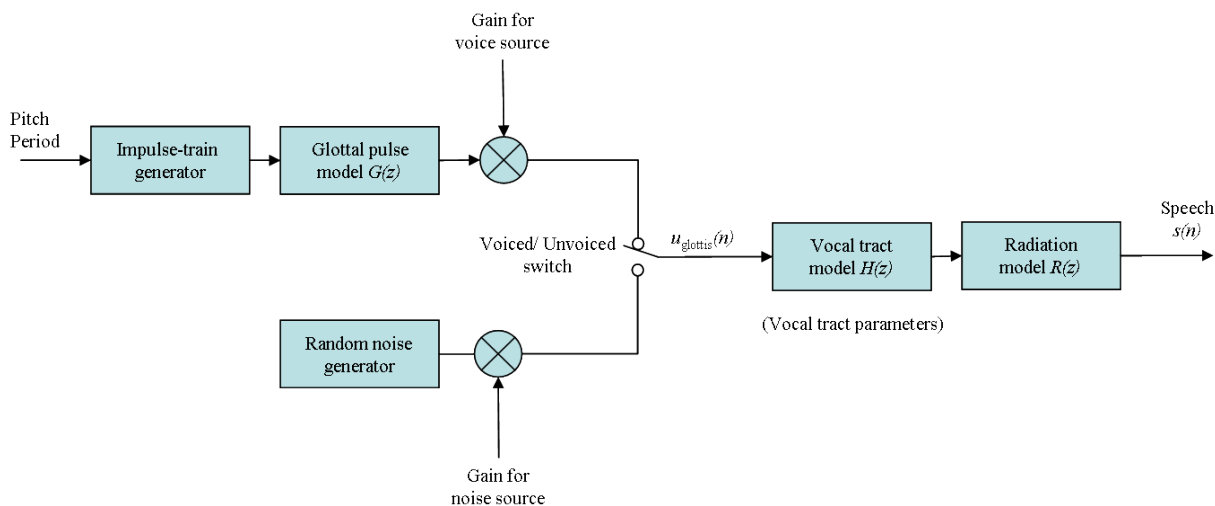


Figure 1.4: General discrete-time filter model for speech production

Creating Acoustic Models

To create acoustic models from speech data, the speech waveform is first subjected to speech preprocessing and feature extraction aimed at optimally obtaining the information in the signal needed for the recognition process. In the case of speech recognition,

information pertaining to the vocal tract movement and the resulting phone sound are both important in the classification of sounds. The differences between speakers, such as pitch and intonation, and external influences of the channel (signal quality or volume) are unimportant and must be filtered out. The feature extraction process windows the continuous speech wave at fixed intervals and codes the spectral characteristics of each window into a vector of real values, called a feature vector. The numerical representations of short segments of speech (typically 10 ms long) can now be used in succession to evaluate speech segments in a digitised environment with statistical modelling techniques. The complete signal processing and feature extraction process is discussed in Chapter 2.

The most popular statistical structure used for acoustic modelling of speech data is a Hidden Markov Model (HMM), due to its ability to model both the overall temporal changes as well as the short-term stationary characteristics of speech sounds. To train the HMM models, labelled speech data that have been accurately aligned with the corresponding model labels are used to collect the feature vectors corresponding to each class. Statistical models are created from these feature vectors by means of parameter estimation. HMM theory, the algorithms used and its application to acoustic modelling are discussed in Chapter 3.

Choice of Subword Modelling Units

In order to decode a speech utterance into a sequence of words, the speech recognition system requires a statistical model for each word in the vocabulary to match to the evaluation sample of speech. For small vocabularies of fixed size it is possible to train statistical models for each word, given that each word has sufficient occurrence in the training data set. For large vocabularies, in excess of 1000 words, this approach becomes impractical due to the excessively large search space for the recognition task and limitations of the training data, which restricts the accuracy and usefulness of each individual model. The system also lacks the ability to adapt and can not handle words that are not in the vocabulary. The problem of scalability is solved by creating statistical models for subword units and combining them to form word models according to a set of lexical rules contained in a lexicon. The subword units form a much smaller set of statistical models that can be used to create models for any number of words in a specific language.

The choice of subword unit greatly influence the complexity, scalability and accuracy of the speech recognition system as we aim to create a system that accurately models the characteristics of human speech within the boundaries of a practical system with limited resources. If the number of statistical models (and therefore statistical parameters) to be estimated are low, the system will have a higher computational efficiency and each subword

unit will occur more frequently in a specific data set (leading to better estimations), but the generality imposed on the system will negatively affect recognition accuracy. If the number of statistical models to be estimated are high, the system will have a lower computational efficiency (often leading to impractical applications), higher demand for resources (e.g. system memory) and not all subword units will occur frequently enough in the data set for accurate estimation. However, the finer modelling capacity will positively affect the recognition accuracy. These considerations are important when choosing a subword unit for a specific speech recognition task.

The smallest set of subword units and consequently the most generalised are phoneme units. Phonemes are modelled irrespective of their position and context within words. Differentiation between the same phone with different preceding phones requires the use of left-context biphones. Conversely, differentiation between the same phone with different succeeding phones requires the use of right-context biphones. Triphones are the set of subword units created by differentiating between both preceding and succeeding phones, simultaneously modelling both the left- and right context. Many commercial speech recognition implementations use pentaphones as subword units, modelling context from the two preceding and two succeeding phones. With larger sets of subword models, adaptation techniques are necessary to limit the need for computational resources and maximise the use of limited training data.

Evaluation of Acoustic Models

Evaluation of a speech recognition system is normally done by noting the word error rate (WER) when the decoded word sequence for a test utterance is compared to the reference labels, obtained through manual transcription. Interim evaluation of the acoustic modelling phase is done by noting phoneme accuracy or the phoneme error rate (*PhER*). Dynamic time warping (DTW) algorithms are used to shorten or lengthen the decoded phoneme sequence in order to minimise the vector distance between the decoded sequence and the correct reference sequence. The number of inserted and deleted phonemes, as well as phonemes that were incorrectly classified are collectively used to calculate the percentage of phoneme errors – the phoneme error rate.

1.3 Literature Synopsis

Through numerous studies on the subject of continuous speech recognition, researchers realised the importance of effective handling of co-articulation in fluent speech. Continuous speech is poorly modelled if it is considered as a concatenated sequence of sounds. A

popular context modelling technique is based on context-dependent models (such as biphones and triphones), but many researchers have also turned to the modelling of phoneme transitions based on diphones as an alternative strategy.

Initial research done on the use of transitional models between 1982 and 1987 focused on in-depth analyses of the transitional effects found in the spectrographic representation of a speech signal. Diphone templates were created to represent each possible phoneme transition and used to compare to unseen speech signals [79, 78, 20, 19, 36].

The multi-trajectory subspace models that evolved from these templates aimed for an efficient representation of each diphone trajectory in a reduced parameter subspace of the original spectrum. Diphones were the subword unit of choice for subspace models because the rich transitional content inherent in diphones could be accurately captured with trajectories. Several studies investigated the potential of diphone-based subspace models [69, 70, 72, 71, 73], doing much to further our understanding of transitional effects in natural speech.

The most popular implementation strategies for modern speech recognition systems involve the use of hidden Markov Models (HMMs) or artificial neural networks (ANNs). Although there are studies using diphones in ANN-based systems [28, 26], the majority make use of HMM-based systems [4, 60]. Three studies in particular proved that diphone-based speech recognition systems achieve higher recognition accuracies than context-dependent models in similar parametric conditions [32, 4, 27].

1. **Fissore et al.** [32] (1996)

This study investigated the problem of defining an acoustic-phonetic unit set for flexible-vocabulary continuous-speech recognition. Aiming to accurately model co-articulation effects in continuous speech while being able to effectively adapt to unseen words, a set of transitional models were used as an alternative to the classic context-dependent modelling approach. Results showed that the system based on transitional units favourably compared to baseline speech recognition systems based on biphones and triphones, if evaluated in similar parametric conditions. System training and evaluation was done on a set of spontaneous sentences recorded through a PBX (6000 training sentences and 858 testing sentences), with the diphone-based recognition system resulting in an average increase in word accuracy of 2.5% over both biphone- and triphone-based systems.

2. **Basztura et al.** [4] (1998)

This study, conducted in Poland, did extensive research into the use of diphones as subword units for Hidden Markov Models-based automatic continuous-speech recognition. An in-depth analysis was done on diphone characteristics, automatic

finding of diphone segments and their parametrisation. Recognition experiments were performed using a hybrid HMM/ANN algorithm on a speech database containing about 115 sentences and resulted in a 9% increase in recognition accuracy in relation to analogous experiments that used phonemes as basic units.

3. **Dobrišek et al.** [27] (1999)

This study directly compared transitional acoustic models with context-dependent phone models by analysing HMM-based recognition systems based on either diphones, biphones or triphones. All systems had approximately the same number of model parameters to enable a direct comparison. Special attention was also given to speech signal segmentation and the effect it has on the eventual recognition systems, concluding that biphone segmentation tends to drift towards diphone segmentation if no initial phoneme alignment is used for accurate biphone labelling. Results showed that diphones achieved higher recognition accuracy than biphones (average increase of 2.2%) and slightly higher than triphones (average increase of 0.2%) on a vocabulary set size of 1000 words and without the aid of a grammar.

This was the framework laid down for the purposes of this research.

1.4 Objectives

The objectives of this research were determined as:

- Evaluation of diphones as effective acoustic modelling units for speech recognition systems, including the selection and application of suitable algorithms used for adaptation and optimisation of the models.
- Implementation of a phoneme recognition system that utilises the finer modelling capacity of diphones by increasing the phoneme recognition accuracy currently achieved by monophone models, while minimising the speed and resource penalties usually experienced with larger model sets.
- Review of subword units commonly used for acoustic modelling by investigating the differences between phone-transition models and more conventional context-dependent models in terms of their ability to model the interphone dynamics and co-articulation effects found in continuous speech.
- Comparison of the performance of diphone-based transition models and biphone-based context-dependent models in terms of complexity, accuracy and computational efficiency in parametrically similar environments.

1.5 Contributions

This research has shown that:

- Diphones are effective subword units that carry suprasegmental knowledge of speech signals, providing an excellent trade-off between proper parameter estimation and detailed co-articulation modelling. Although the advantages of using diphones have been established in the field of speech synthesis, research related to diphone use in speech recognition has been limited and more often applied to the use of non-parametric modelling techniques. This research proved the value of a diphone-based speech recognition system used within the framework of Hidden Markov Model theory.
- With the use of adaptation techniques transition modelling through the use of diphone-based acoustic models can increase recognition accuracy while retaining computational efficiency.
- In a system with limited resources where computational efficiency is important, diphone models outperform monophones and biphones in terms of phoneme recognition accuracy, without the use of language models and/or grammar.

1.6 Thesis Overview

This chapter provides a broad overview of research done for this thesis. The motivation (Section 1.1) for embarking on this research topic is supported by a short literature synopsis (Section 1.3) and basic background theory explaining key concepts pertaining to acoustic modelling and speech recognition (Section 1.2). The description of the project is brought full circle with a list of what we wished to accomplish with this research (Section 1.4) and a summary of contributions made in the process (Section 1.5). The rest of this thesis can be viewed in the light of these categories, which are summarised in the following subsections:

1.6.1 Background Theory on Statistical Speech Recognition

In Chapters 2 and 3 the theory of statistical speech recognition and acoustic modelling is explained. These chapters have the dual purpose of putting the research done for this thesis into perspective with respect to the larger context of the field of speech recognition and to give readers who are not familiar with the field the necessary background information to be able to understand the concepts discussed in the rest of the thesis.

Chapter 2 starts with a general overview of the speech recognition problem. A short history of research done the field of speech recognition and the changes in methodologies over the years are reviewed in Section 2.2. This is followed by a discussion on the use of diphones in speech recognition with reference to specific studies directly related to this research. The chapter concludes with the mathematical formulation that underpins speech recognition and descriptions of the various components found in the speech recognition system (Section 2.3). These include digital signal processing used to extract feature vectors from the speech signal, acoustic modelling, including the important consideration of which subword unit to use, lexical modelling and language modelling.

Chapter 3 contains an extensive discussion on hidden Markov models (HMMs) and their use in speech recognition applications. A detailed mathematical description of HMM theory (Section 3.1) and the algorithms used to train and evaluate them (Section 3.2) are provided, as well as a discussion on the integration of HMM theory into speech recognition applications (Section 3.3). This chapter culminates in a discussion of the way knowledge of the mathematical base of HMM theory and its use for speech recognition are applied in acoustic model training for phoneme recognition using HMM models (Section 3.4), which form the base of all experiments done in this research. The most important aspects of acoustic modelling are the accurate alignment of the labelled training set, the design of each HMM (topology and output probability distribution functions), model training, decoding of a speech signal using the models and evaluation of the results. These issues are discussed in context of phoneme recognition as done in this research.

1.6.2 Analyses of Diphones and their Use in Speech Recognition

The goal of this research is the objective analysis of diphones as subword units in continuous speech recognition. It is therefore necessary to define a diphone and its relation to other candidates that can be used as subword units.

Chapter 4 contains a thorough examination of diphones and their use in speech recognition applications. Diphones are first put into perspective by comparing them with the most popular subword units used in speech recognition, such as syllables, monophones, biphones and triphones (Section 4.1). These subword units can be classified as either context-independent (CI) or context-dependent (CD), depending on whether they are designed to incorporate contextual information into the acoustic model or not. Context-dependent modelling is designed to handle the co-articulation effects found in continuous speech which can have a significant impact on the recognition system accuracy. Diphones are a special case of context-dependent models, called transitional models, as they are designed to model the transitions between subsequent phonemes instead of focusing on the

phonemes themselves. Biphones are context-dependent models closely related to diphones and serve as an excellent base for the comparison of transitional models and context-dependent models, based on various criteria. These criteria are discussed in Section 4.2 and include trainability in terms of data scarcity and the robustness of segment boundaries, complexity, resource requirements, handling of inter-word contexts and modelling of unseen contexts. For the most part, both context-dependent models and transitional models share the same problems, such as sensitivity to data scarcity, but there is one criterion that favours transitional models – robustness of segment boundaries. The segmentation of a speech signal for transitional models places the segment boundaries in the relatively stationary portion in the centre of each phoneme, whereas segmentation based on the start and end of phonemes means that boundaries are placed in a fast changing portion of the signal, therefore small changes in boundary locations have a large influence on the acoustic model training.

As an extension of the literature study done in Section 2.2, different implementation strategies for using diphones in speech recognition are briefly explained in Section 4.3. This section aims to bring together what we know about acoustic modelling, speech recognition theory and diphone characteristics. A lot can be learned about diphones from the different strategies employed in using them for speech recognition. These implementation strategies include non-parametric methods, such as template extraction and multi-trajectory subspace models, and parametric methods, such as neural networks and hidden Markov models. Automatic segmentation of the speech signal into diphone units is also discussed, including a technique borrowed from the field of speech synthesis to align the data with the output from a synthesised utterance of the transcription.

The diphone study concludes with a discussion on the use of diphones in HMM-based systems in Section 4.4. Specific attention is paid to segmentation, model structure and the adaptation of the decoding and evaluation methods discussed in Section 3.4 for use with diphones.

1.6.3 Experiments, Results and Conclusions

Chapter 5 details the implementation of various adaptation techniques used to improve the performance of the diphone-based recognition system and Chapter 6 contains all experiments designed to evaluate the diphone-based recognition systems based on these adaptation techniques and their performance relative to a monophone-based baseline system and a context-dependent biphone-based system.

The first technique is diphthong splitting, used to divide “double phonemes” (two phonemes in quick succession usually considered a single phoneme) into their constituent

phonemes, each of which is already contained in the phoneme set. Diphones explicitly model transitions, therefore including diphthong phonemes in the system constitutes redundancy. Diphthong splitting is explained in Section 5.1 and evaluated in Section 6.3.2. Another technique used to exploit the characteristics inherent in diphones is implementing a basic diphone grammar for decoding. The diphone structure places restrictions on which diphones can follow a specific one, increasing decoder complexity as well as accuracy (Section 5.2). The basic diphone spotter was used in most diphone experiments detailed in Chapter 6, with a few experiments designed to isolate its influence on the recognition system.

The biggest challenge for a diphone-based recognition system is in the effective handling of limited training data due to the very high class set size and the subsequent low class representation, leading to poorly estimated model parameters. To address this issue, diphones were built from well-estimated monophone models and used as prior estimations for maximum a posteriori (MAP) estimation, which resulted in an increase in recognition accuracy over a monophone-based baseline system at the cost of a significant increase in the execution time of phoneme decoding, due in part to limitations of the current hardware system. MAP estimation is explained in Section 5.4 and evaluated in Section 6.3.3.

Decision-tree based state clustering through the use of CART trees is an effective technique that can be used to group the output probability density functions on the HMM states in such a way as to increase class representation in the data set, lower resource requirements and increase parameter reliability. This is a technique often used when utilising large sets of acoustic models with high complexity such as context-dependent models. The resulting system is much more efficient and accurate. Decision-tree based state clustering is explained in Section 5.5 and its use in the diphone-based system is evaluated by means of various experiments in Section 6.3.5. The best diphone-based system was obtained using the well-estimated diphone-adapted models obtained after MAP estimation to re-align the training set as a base for the clustering procedure, reducing the total number of output probability densities to roughly 20% of the original size. The phoneme accuracy of the best diphone-based system is about 8% (absolute) better than the equivalent monophone-based system and decodes approximately 4 times slower than the monophone-based system, which is an acceptable cost for the gain in accuracy.

The diphone-based system that performed the best on the platform used for the experiments is ultimately compared to a biphone-based system in Section 6.4 in similar parametric conditions. The biphone-based system utilises biphone models adapted to use the same techniques as the best diphone-based system. To provide firmer, more reliable biphone boundaries, a slight bias in favour of the biphone models was introduced, by

basing them on an improved monophone set. Despite this bias, the diphone-based system outperformed the biphone-based system with an increased phoneme recognition accuracy margin of approximately 2%.

Chapter 2

Speech Recognition: Theoretical Background

The field of human language technology comprises a range of activities, with the objective to enable communication between humans and machines using natural language. Research includes recognition, decoding and interpretation of human-produced speech signals at one end and the production of speech signals at the other. These two broad classifications are commonly referred to as speech recognition and speech synthesis. They can be used individually for applications such as voice-enabled dialling, data mining in audio signals and voiced warning systems, or they can be linked and combined with artificial intelligence to create applications such as a conversational avatar.

Speech recognition can be further divided into sub-disciplines according to the final goal, which varies from determining the linguistic content in the speech signal (continuous-speech recognition), the identity of the speaker from a set of known speakers (speaker recognition), the language used (spoken language recognition) and whether or not a specific person is speaking (speaker verification). The work described in this thesis is aimed at continuous-speech recognition applications, but because it is more closely tied to acoustic models than linguistic models, it can be adapted for use with other types of speech technology applications as well.

2.1 Types of Speech Recognition

Speech signals are composed of a sequence of sounds that serve as symbolic representations for the thoughts the speaker wishes to convey to the listener. These sounds interact and combine to form words associated with a language from which the meaning is extracted by the listener. Speech recognition is the process of converting the acoustic speech signal,

captured by a microphone or a telephone, to a sequence of words through the use of acoustic and language modelling techniques.

Speech recognition systems have many parameters that characterise them depending on both the application and the available speech corpus used. This makes direct comparison of different systems very difficult. It is therefore important to define the problem to be solved, and especially the input data, associated with any research.

Isolated-speech Recognition and Continuous-speech Recognition

Isolated speech recognition is the recognition of words spoken in isolation with well-defined pauses between them. It is usually used with a small vocabulary, representing spoken commands or simple voiced data entry for specific applications, including voice dialling, call routing and technological assistance. It generally has a high accuracy. Continuous-speech recognition is the decoding of natural speech utterances, which is vastly more complex. A much larger vocabulary is used and it is highly influenced by co-articulation, resulting in lower recognition accuracies.

Read Speech and Spontaneous Speech

Speech that is being read from a script has a predefined structure, resulting in a cleaner sequence of words, easily accessible transcriptions and a fixed vocabulary size. However, read speech can sound mechanical with fluctuations in tone that are not present in spontaneous speech. Acoustic models trained solely on read speech perform poorly when used to evaluate natural speech data because of the prosodic differences between the two [58]. Spontaneous speech is usually not fluent – there are frequent false starts, pauses and mid-sentence breaks. Sounds such as coughing or laughing are almost always present, especially when the speech data contains dialogue, and often words are used that are not in the vocabulary. Additionally, transcriptions have to be generated manually for all training data, which is a costly and time-consuming task. Although spontaneous speech contains a more natural tone, it is very hard to work with, resulting in lower recognition accuracies.

Speaker-dependent and Speaker-independent Speech Recognition

In speaker-dependent recognition systems, acoustic models are trained on speech data provided by one speaker. These systems require less training data to sufficiently represent the signal characteristics, but their usage is limited. Speaker-independent speech recognition systems are general systems that do not require a user to record training utterances before it can be used. These systems generally have a lower recognition accuracy

and require a larger amount of training data to prepare the acoustic models for a wide variety of voice types, speaking styles and accents.

Vocabulary Size

The vocabulary size directly influences the complexity of speech recognition, with small vocabularies (less than 20 words) achieving high accuracies compared to large vocabularies (in excess of 20,000 words). Applications that have limited input variance are easy to implement and use, but are of little use in real-world situations. Large-vocabulary speech recognition often require the use of complex grammatical models to assist in generating logical word sequences, but languages constantly evolve and grow, making handling of out-of-vocabulary words essential.

Language Model

Speech recognition often does not end with phoneme or word recognition. Lexical and grammatical decoding of the underlying subword unit sequence lead to sentence construction and ultimately derivation of the intent of the speaker. Complex language models are used to restrict the possible word sequences that can be recognised from a given speech sample. These language models can either be statistically derived from a large amount of speech or written data, or they can be implemented as a set of linguistic rules.

Input Signal Recording and Handling

Speech that has been recorded in a studio under controlled conditions yields better quality speech models and consequently better recognition results. However, gathering microphone-recorded speech data is a slow and expensive process making it impractical for most applications.

To use the speech signal in a digitised environment it is sampled at a specific frequency, typically 10kHz or 16kHz for microphone-recorded speech and 8kHz for telephone-recorded speech. The lower sampling frequency for telephone-recorded speech is due to the limited bandwidth used to transmit the data over a telephone line. A standard land-line telephone has a maximum transmission bandwidth of 64 kilobits per second (8000 samples per second multiplied by eight bits needed to represent each sample). The quality of speech data transmitted over a telephone line is therefore significantly lower than for microphone-recorded speech. Speech recorded over a telephone is often used for modern speech recognition systems because the process is fast, utilising a large set of speakers with readily available equipment. The drawback of using telephone speech is the low quality

of the signal. The noise content is high, requiring noise-cancellation algorithms. As can be expected any adverse conditions, such as noise, signal distortion and transmission line variability will drastically degrade the system performance.

The Experimental Setup

The research done for this thesis pertains to acoustic modelling for speaker-independent continuous-speech recognition trained on a mixture of scripted and spontaneous telephone-recorded speech for use in large-vocabulary speech recognition. Experiments were done on the AST (African Speech Technology) data set of English speech for native- and non-native English speakers collected in South Africa. A detailed discussion of the speech corpus can be found in Appendix B.

This research setup is consistent with the trend of the past two decades where the significant progress already made in basic speech recognition technology lead researchers to tackle more difficult but more practical problems. In laboratory experiments it is not uncommon to encounter word recognition accuracies as low as 50% for speaker-independent continuous-speech recognition trained on telephone-recorded speech. Commercial products utilising complex systems with state-of-the-art technology can achieve word recognition accuracies of up to 99% [21].

2.2 Literature Study

2.2.1 A Brief History

The scientific community was first introduced to the notion of combining knowledge from the fields of linguistics and computer science when Warren Weaver wrote his famous memorandum in 1949 suggesting that translation by machine may be possible. Warren Weaver's vision of machine translation originated in his war-time experiences as a cryptographer when he started pondering the use of statistical methods to derive a characterisation of translation from textual input. The founders of computational linguistics were, however, not statisticians but linguists and they saw the potential of the computer not in statistical analysis of large amounts of data, but rather in carrying out minutely specified rules that they would write. This inference of knowledge, based on the notion that human communication is a deductive system that can be reduced to a complete set of rules, was fuelled by prominent scientists such as Chomsky (1957 - Syntactic Structures [17]) and a general distrust in the use of statistics to increase our understanding.

The years between 1950 and 1970 can be seen as the pioneering era for speech research

characterised by interdisciplinary contacts. The sound spectrograph, developed at Bell Laboratories in the 1940s, provided insights into the nature of speech signals and their relationships to the linguistic frame. An early acoustic-phonetic study of spectrographic data, feature theory and the temporal distribution of information bearing elements appeared in [31] to detail progress made by researchers at KTH – Royal institute of technology in Sweden in 1962. In the early years, computational linguistics was dominated by linguistic theory: finding generalised phrase structures and lexical functional grammars and applying finite-state methods to phonology (study of sound patterns in a language) and morphology (study of word structure). Limited success with this approach led to the realisation that human communication is not just a set of rules, but that our intelligence is integral in the decoding of meaning, especially when the information gathered is incomplete and informal. Humans have the ability to apply knowledge not only of the current situation, but using information gathered in a larger context to resolve ambiguity and extract meaning in a conversation. Linguists were forced to find another scientific framework to embed their systems in and found it in statistics. At the same time engineers were working on acoustic-phonetic and feature theory, intelligibility in speech signals and speech compression for bandwidth reduction.

With the development and availability of the microcomputer in the early 1970s, which provided computing and storage capacities comparable to the previously dominant mainframe computers, calculations could be done in a fraction of the time at much lower cost than previously possible. Computers were becoming standard laboratory tools, marking the transition from analogue to digital processing in all aspects of speech research. Human-machine interaction was rethought, leading to a growing interest in computational linguistics. Between 1970 and 1985 major advances were made in statistical modelling techniques by influential scientists such as Baum [7], Levinson [49], Rabiner [67, 65], Bahl [2] and Jelinek [44, 3, 57]. Stochastic approaches in speech modelling were preferred above deterministic template-matching because of the ability to inherently characterise the variability in speech. The prevalent theories at the time were either related to non-parametric methods such as using nearest-neighbour type algorithms for sample comparison to identify the most probable classification, or stochastic modelling techniques. In contrast with non-parametric methods, stochastic modelling is used to derive a parametric model for each word in the vocabulary, with an associated likelihood function that is used to determine the probability that the unknown point represents an instance of the current word. These methods provided reasonable word recognition accuracies for independent-speaker recognition of small vocabulary data sets [41].

Statistically oriented methods of speech processing such as Hidden Markov Models

(HMMs) and Artificial Neural Networks (ANNs) became standard tools for use in speech modelling applications between 1980 and 1995. This era also saw a movement towards the more difficult problems of speech recognition such as large-vocabulary continuous-speech recognition in less than perfect conditions. Theoretical advances in the laboratory were shifted to research and development of applications in commercial speech technology products. Since 1995 the concept of automatic speech translation over a telephone line has become a topic of great interest for researchers, as it combines virtually all information relating to speech technology – speech and language recognition, analysis of speaker type, speaking style and emotions, language translation and re-synthesis of speaker characteristics and speaking style – adjusted to match the demands of the target language.

Although statistics has provided a reliable mathematical base for speech recognition research, there is a growing insight that we cannot realize far-reaching goals without incorporating more fundamental knowledge about human speech communication and information-bearing elements of speech [30]. Statistical speech recognition is therefore the intersection of linguistic theory and statistics theory – both bring valuable contributions to the field.

2.2.2 The Use of Diphones in Speech Recognition

Early analysis of speech signals did much to further our understanding of how human communication works. Spectrographic and temporal analysis revealed the intricate nature of continuous speech patterns, especially during transitions from one sound to another. These transitions proved to be very important for speech perception and was quickly adopted as the subword unit of choice for speech synthesis (specifically concatenative synthesis attempting to generate synthesised speech by concatenating acoustic subword units) in the form of diphones or demisyllables [55].

An overview of important research on diphone segments as used in speech recognition is given below, which is by no means a complete account. Over the years researchers exploring the field of speech technology have used transitional segments such as diphones in experiments that were not necessarily designed to investigate diphones as subword units for speech recognition. In the research detailed below diphones or diphone-like units were chosen, either because they are very well suited to a specific speech recognition methodology or because researchers wanted to investigate the advantages of using diphones rather than another subword unit. System accuracy is denoted using the word error rate (WER) or word recognition rate (WRR), unless otherwise specified.

Diphone Templates

Diphones were first used as base subword units in speech recognition research in the 1980s. Knowledge of the transitional characteristics inherent in diphone units were used to create dictionaries of acoustic diphone templates [79, 78]. Existing pattern matching algorithms could then be used to dynamically match an unknown input signal to the diphone templates to find the most likely diphone sequence. The use of diphone templates for speech recognition is an example of a non-parametric statistical method that considers input data without attempting to fit a parametric mathematical or statistical function onto the data.

Experiments on a small-vocabulary speech data set, such as a spoken sequence of digits, allowed researchers to closely analyse the transitional characteristics of a small set of diphones and manually create templates that reflect spectral variations over time. Preliminary experiments showed great promise, but in order to use diphone templates in larger vocabulary environments, algorithms that could automatically bootstrap the diphone templates were needed [20, 19]. Three prevalent studies are summarised in Table 2.1.

Table 2.1: *Summary of research into the use of **diphone templates** for speech recognition*

Year	Author	Ref	Type of speech recognition	Data set	Accuracy
1982	Scagliola <i>et al.</i>	[79]	Speaker-dependent; Small-vocabulary; Continuous-speech	Connected digit sequences	98% on sequences
1984	Colla <i>et al.</i>	[20]	Speaker-adaptive; Small-vocabulary; Continuous-speech	Short Connected digit sequences; Small data set	94.48% on sequences; 98.79% WRR
1987	Colla <i>et al.</i>	[19]	Speaker-adaptive; Small-vocabulary; Continuous-speech	Connected digit sequences; Larger data set	96% avg WRR

During the evolution of research in the field of speech recognition, the most popular and influential implementation strategy was the hidden Markov model (HMM). Hidden Markov models are simple, yet effective statistical models that have the ability to model both dynamic temporal changes and short periods of stationarity present in speech signals. One area of research investigated the integration of HMM theory with current well-known systems such as speech templates. A study in 1993 used a piecewise-linear set of templates

as representations of a typical sequence of observations within an HMM framework [36]. The HMM/template hybrid system combines the detailed modelling capabilities of the templates with the solid mathematical framework of an HMM system for reliable model training and parameter estimation. The study used diphones as basic subword units because of their ability to accurately model co-articulation. Diphones are also the smallest subword unit for which templates are ideal because of the rich data content and high rate of change within the transitions. The study showed that the diphone HMM/template hybrid system can achieve significantly higher recognition accuracies than a baseline HMM system.

Table 2.2: *Summary of research into the use of **hybrid HMM/template systems** for speech recognition*

Year	Author	Ref	Type of speech recognition	Data set	Accuracy
1993	Ghitza <i>et al.</i>	[36]	Speaker-dependent; Medium-vocabulary; Continuous-speech	Sentences	avg 15% reduction in phoneme errors over baseline HMM systems

Subspace Models

Subspace models are similar to templates in that they also provide temporal descriptions of a segment to characterise the changes inherent in the speech signal over time. The difference is that subspace models aim to optimally reduce the dimensionality of the feature vector space while still preserving the temporal ordering of the data sequence. This technique is known as time-constrained Principal Component Analysis (TC-PCA), which is derived from the algorithm for finding *Principal Curves* (first described in 1989) [69]. Subspace models are able to finely model the non-linear transient characteristics contained in diphones. Extensive research has been done on the subject of subspace models with diphones as base units by researchers K Reinhard and M Niranjana [69, 70, 72, 71].

The accuracies achieved were slightly lower than baseline HMM-based systems, but the subspace models contained relatively few parameters in comparison to conventional hidden Markov models. Subspace models of diphone units were subsequently used in combination with HMM systems [73] to improve overall recognition accuracies.

Table 2.3: *Summary of research into the use of **diphone subspace models** for speech recognition*

Year	Author	Ref	Type of speech recognition	Data set	Accuracy
1998	Reinhard; Niranjan	[70]	Speaker-independent; Small-vocabulary; Isolated-speech	Isolated spoken characters	78.3% diphone recognition accuracy
1999	Reinhard; Niranjan	[71]	Speaker-independent; Small-vocabulary; Continuous-speech	TIMIT (diphone subset)	avg 50% diphone recognition accuracy
2000	Reinhard; Niranjan	[73]	Speaker-independent; Small-vocabulary; Continuous-speech	TIMIT (subset)	4.6% relative improvement over baseline HMM systems

Neural Networks

An artificial neural network (ANN) is a mathematical model designed to create artificial intelligence by directly simulating neurological functioning in the human brain. The model consists of layers of interconnected nodes (“neurons”), each containing a mathematical function for determining whether or not the signals received from the node inputs are to be propagated (“neuron firing”) onto the node outputs. The concept of neural networks were first conceived in the 1940s, but it wasn’t until the development of an efficient training algorithm for multi-layer perceptron networks in 1986 that neural networks could be considered for speech recognition problems [77].

An ANN is not designed to model temporal variation and is therefore usually used in conjunction with HMM-based systems. The HMM structure provides a piecewise-linear framework for characterising short segments of the speech signal with the aid of a neural network. Two studies by Belgian researchers in 1997 [28] and 1999 [26] investigated the use of hybrid HMM/ANN systems for medium to large-vocabulary speech recognition applications using diphones as basic subword units. Results showed a slight improvement in recognition accuracy over baseline HMM systems, although the HMM/ANN hybrid systems are more complex and more resource intensive than baseline HMM systems. In another study done in Poland in 1998, a diphone-based HMM/ANN system outperformed a similar HMM/ANN system based on phonetic subword units [4].

Another study, done in Korea in 1996, used a time-delay neural network (TDNN)-

based diphone recognition system for spoken Korean [47]. Diphones are preferred subword units for speech recognition in Korean because Korean phonemes are all very similar to each other. Diphones therefore result in a more discriminant set of subword classes that also has the advantage of being able to model contextual information.

Table 2.4: *Summary of research into the use **HMM/ANN hybrid systems** for speech recognition*

Year	Author	Ref	Type of speech recognition	Data set	Accuracy
1996	Lee & Lee	[47]	Speaker-dependent; Medium-vocabulary; Continuous-speech	Utterances	80.6% correctness in morphological analysis
1997	Dupont <i>et al.</i>	[28]	Speaker-independent; Medium-vocabulary; Isolated-speech	Isolated words	5% decrease in WER from similar HMM systems
1998	Basztura <i>et al.</i>	[28]	Speaker-independent; Medium-vocabulary; Continuous-speech	Sentences	9% decrease in WER from similar monophone- based systems
1999	Deroo <i>et al.</i>	[26]	Speaker-independent; Large-vocabulary; Isolated-speech	Isolated words	avg 6% decrease in WER from similar HMM systems

Hidden Markov Models

The most prominent parametric implementation strategy for statistical speech recognition is the use of hidden Markov models. Numerous studies have shown the advantages of HMM based recognition systems – a solid mathematical framework, reliable model estimation algorithms, simplicity and efficient use of limited system resources. HMM theory is discussed in detail in Chapter 3.

Several studies aimed to investigate the use of diphones or diphone-like units within the framework of HMM-based recognition systems. As these studies are directly in line

with the objectives of this research, they are detailed below.

- **1996** – **Fissore** *et al.* [32]

As an alternative to context-dependent modelling techniques using biphones and triphones, a set of transitional state units is defined for use in a flexible-vocabulary continuous-speech recognition system. Linear 3-state left-to-right hidden Markov models are trained for each diphone with Gaussian mixture densities as state output probability densities. The speech corpus consists of a set of spontaneous sentences recorded through a PBX (6000 training sentences and 858 testing sentences). The recognition system based on diphone units scored a 2.5% average increase in word accuracy over a similar system based on context-dependent biphone and triphone units. These initial results showed that diphones can achieve high recognition accuracies with little additional requirement of computational load.

- **1998** – **Mariño** *et al.* [54]

The use of demiphones as subword units is directly compared to similar systems using context-dependent triphone units. Demiphones are defined as subword units that independently model the left and right halves of a phoneme in context with neighbouring phonemes. Demiphones can therefore be seen as two halves of a diphone or triphone. Continuous density hidden Markov models are used together with decision-tree based state clustering to optimally reduce the number of parameters. The study concluded that demiphones simplify the speech recognition system and yield better recognition performance than similar systems based on triphones. Demiphones also provide an excellent tradeoff between detailed co-articulation modelling and proper parameter estimation.

- **1998** – **O’Neill** *et al.* [60]

Many studies in speech recognition recognise the value of explicitly modelling contextual information in speech signals. In this study a particular case of multi-phone strings, namely phone-pairs is investigated in detail to find the balance between longer duration models with better context modelling and the simplicity and generality of smaller subword units, such as phonemes. Phone-pairs differ slightly from diphones as they model two whole phonemes instead of modelling only the transition between the two. Phone-pairs were modelled using 6-state left-to-right HMMs with Gaussian mixture densities. On average the recognition system achieved a 10% higher recognition accuracy when using phone-pairs than with left-context biphones. It was concluded that phone-pairs are an effective alternative to context-dependent models for modelling contextual information.

- **1998 – Glass & Hazen [38]**

JUPITER is a telephone-based weather information system, which is available via a toll-free number for users to query a database of current weather conditions using natural, conversational speech. The JUPITER system makes use of the GALAXY conversational system architecture, which incorporates speech recognition, language understanding, discourse modelling and language generation. New data is continually collected to incorporate into the system and improve acoustic modelling and overall functionality. This impressive system makes use of diphones as subword models. The maximum word error rate (WER) achieved in 1998 for standard users was 25% (children) and the minimum was 8.4% (adult males).

- **1999 – Dobrišek *et al.* [27]**

This study specifically focused on comparing diphone models with biphone models – the more traditional context-dependent equivalent. The study emphasised the importance of accurate speech signal segmentation and investigated how it affects diphone and biphone model accuracy. In experiments where models were trained without proper segmentation, diphones and biphones produce similar results. Decision-tree based state clustering was used to reduce the total number of system parameters of both diphones and biphones to similar values, to enable direct comparison. Results showed that when a bigram grammar was used, diphones (95.8% accuracy) performed better than both biphones (92.5% accuracy) and triphones (93.9%), which was a promising sign for transitional models.

These studies proved the potential of transitional models, especially those based on diphones as subword units for speech recognition systems.

2.3 The Speech Recognition System

The goal of speech recognition is to find the most likely word sequence given an observed acoustic signal. The system is evaluated by noting the word error rate (WER) of the final output. Interim results such as phoneme recognition accuracies can also be used to evaluate the effectiveness of the acoustic models without the use of a language model.

2.3.1 Mathematical Formulation

Let \mathbf{X} denote a sequence of acoustic features derived from the observed signal,

$$\mathbf{X} = x_1, x_2, \dots, x_m \tag{2.1}$$

and \mathbf{W} denote a subset of n words belonging to a fixed and known vocabulary.

$$\mathbf{W} = w_1, w_2, \dots, w_n \quad (2.2)$$

Then the probability of a specific word sequence \mathbf{W} given the observed features \mathbf{X} is denoted as $P(\mathbf{W}|\mathbf{X})$. The most likely word sequence is found by iterating over all possible word sequences in the vocabulary and finding the maximum likelihood probability, or

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} P(\mathbf{W}|\mathbf{X}) \quad (2.3)$$

Using Bayes' formula of probability theory, it can be rewritten as

$$P(\mathbf{W}|\mathbf{X}) = \frac{P(\mathbf{X}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{X})} \quad (2.4)$$

where $P(\mathbf{X}|\mathbf{W})$ is the probability that feature sequence \mathbf{X} is observed given that we know that word sequence \mathbf{W} was spoken. $P(\mathbf{W})$ is the probability that the word sequence \mathbf{W} will occur and $P(\mathbf{X})$ is the probability that the sequence of features \mathbf{X} was observed. The problem of finding the most likely word sequence given the fixed observation sequence \mathbf{X} reduces to finding the word sequence $\hat{\mathbf{W}}$ that maximises the probability

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} P(\mathbf{X}|\mathbf{W})P(\mathbf{W}) \quad (2.5)$$

From Equation 2.5 the components necessary for speech recognition can be identified. An acoustic processor is needed to extract the sequence of acoustic features \mathbf{X} , from the observed signal. To calculate $P(\mathbf{X}|\mathbf{W})$, statistical models are needed to calculate the probability that the speech recognition system produced the acoustic features \mathbf{X} given that we know that word sequence \mathbf{W} was spoken and therefore know the configuration of the acoustic model set. These statistical models are created during acoustic modelling and are used to evaluate unknown speech data through acoustic decoding. To calculate $P(\mathbf{W})$, the probability that the speaker uttered the word sequence \mathbf{W} , language models are needed to evaluate the validity of the word sequence. Lexical modelling and language modelling are performed not only on the speech data used for acoustic model training, but also on a large amount of external training data used solely for the purpose of morphological and syntactic modelling.

2.3.2 Components of the Speech Recognition System

Statistical speech recognition has two distinct phases: speech modelling (training) and speech decoding (recognition). Speech modelling refers to the training of statistical models on a training data set and speech decoding is the evaluation of new data by utilising the

trained statistical models. The statistical models are parametric representations of the reference patterns for the different speech sounds.

The first component of the speech recognition system is common to both these phases – digital signal processing and feature extraction, used to convert the speech signal into a time-sequence of vectors that can be used by computer software. During speech modelling acoustic, lexical and language modelling are used to create the necessary statistical models to represent both the acoustic features of speech sounds and their interactions to form word sequences. Speech decoding utilises these statistical models in turn to perform acoustic, lexical and language decoding. To measure the effectiveness of the system a testing data set for which manual transcriptions exist are used to compare the decoded system output to the correctly labelled output.

Figure 2.1 shows these components for both the modelling and decoding phases. These components are discussed in Sections 2.3.3 to 2.3.6. Note that the focus of this research is limited to the components contained within the block marked with dashed lines, which will be discussed in detail in the following chapters.

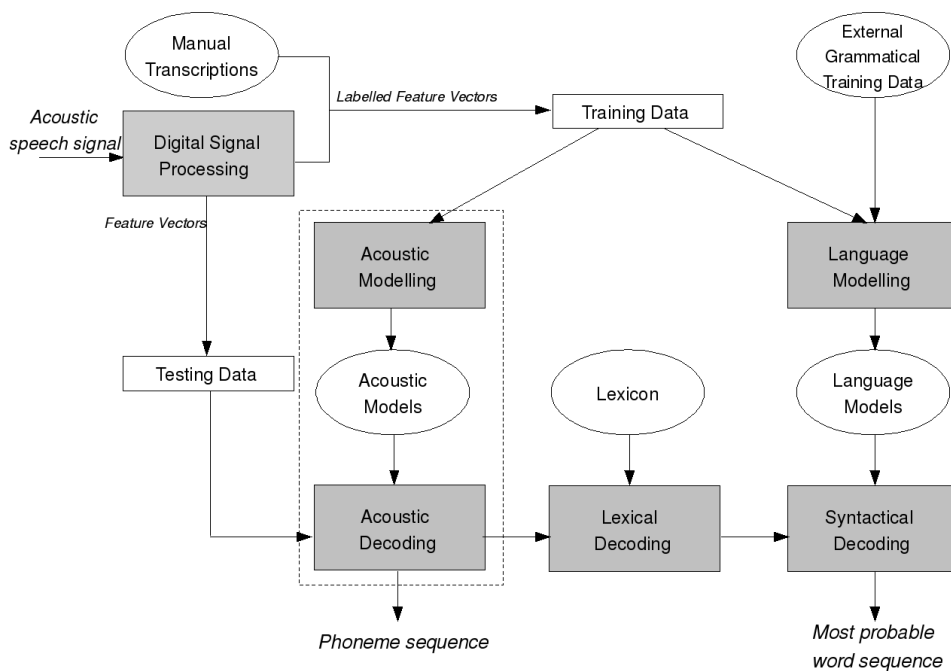


Figure 2.1: System diagram of the basic components in a speech recognition system

2.3.3 Digital Signal Processing for Speech Signals

For speech recognition, *Digital Signal Processing* (DSP) techniques are used in place of the biological system humans use for speech perception to extract information from the speech signal. DSP on speech signals for use in speech recognition aims to optimise the extraction of valuable information that can be used in pattern recognition. The representation of the data in feature vectors aims to preserve the information contained in the speech signal that can be used to characterise the identity of a portion of speech, while minimising the effects of other factors such as external noise, speaker differences and even the emotional state of the speaker. The extracted information also needs to be presented in a compact form in order to minimise the need for computational resources.

The main components of the process are signal preprocessing, feature extraction and feature post-processing.

Signal Preprocessing

Signal preprocessing is an important step in effective feature extraction of speech signals. Its purpose is to minimise the effect of all unwanted external influences on the signal to better isolate the information that will be modelled. For speaker-independent speech recognition the important information is the vocal tract movement and the resulting sounds that characterise a phoneme. The differences between speakers (pitch, intonation) and external influences of the channel (signal quality, volume) need to be normalised to provide a space where different speech signals can be compared. The two most important preprocessing techniques are pre-emphasis and power normalisation.

Pre-emphasis filters are first-order high-pass filters used to enhance the relative energy of high-frequency components in the spectrum while attenuating low frequencies, making the average spectrum roughly flat. This is necessary because the amount of energy present in human speech decreases as the frequency increases. The frequency response of the pre-emphasis filter typically takes the form of a single-zero filter

$$P_r(z) = 1 - \nu z^{-1}, \quad \nu \approx 1.0 \quad (2.6)$$

where ν usually has a value of between 0.9 and 1.0, although the precise value seems to be of little consequence [24]. This filter results in a 6-dB per octave shift on the speech spectrum, eliminating the spectral contributions of the lip radiation.

Power normalisation is used to normalise average power in different speech recordings caused by differences in the recording process. Without it, the features extracted from signals with differing average power for a specific speech sound cannot be compared.

Given a speech signal $s(n)$, the average power of the signal is

$$P_s = \frac{1}{L} \sum_{n=1}^L s^2(n) \quad (2.7)$$

where L is the sample-length of the signal. The power-normalised signal is calculated as

$$s'(n) = \frac{s(n)}{\sqrt{P_s}} \quad (2.8)$$

This normalisation process is often done sequentially on smaller subsets of the signal rather than on the signal as a whole. By computing the power over the entire signal, segments of particularly high energy such as loud noises can dominate the calculation of power in the signal. This can lead to negative effects on the rest of the speech signal, especially segments of low energy, for example silence. The isolation and normalisation of power information in shorter segments lead to better feature extraction from the short-term spectrum.

Feature Extraction

In order to use an acoustic signal in a digitised environment such as a computer, the continuous signal must be converted to discrete samples. An inherent assumption is that although the speech signal continually changes with time, it can be broken down into a series of short segments that are quasi-stationary for short periods (between 10ms and 20ms) due to physical limitations on the speed of articulatory movement. The sampling process therefore does not discard information if the sampling rate is sufficiently high.

After initial sampling the next step, called feature extraction, aims to extract the most important linguistic information that will be used to distinguish between different sounds or phonemes. The sampled speech is divided into sets of consecutive frames by applying a window such as a Hamming window, which minimises the amount of spectral leakage due to discontinuities caused by windowing the signal. The windows usually overlap and create vectors that are spaced between 10 and 20 ms apart.

These sets are then processed to extract parameters (features) that characterise the frames, producing a sequence of vectors (the feature vectors). Each feature vector is a D -dimensional vector of real values, which maps the speech data to a point in a D -dimensional space.

Two main types of parameter extraction algorithms are used in speech recognition applications today. One is based on the short-term Fourier spectrum of the frame sets and the other is based on *Linear Predictive Analysis* (LPA). The feature extraction process applied in this research uses a cepstral analysis technique, which is based on analysis of

the Fourier spectrum. LPA falls outside the scope of this research and will therefore not be discussed any further.

Representations used in current speech recognisers concentrate on properties of the speech signal attributable to the shape of the vocal tract rather than on the excitation of the signal. The excitation of the speech signal occurs when air is forced from the lungs and through the vocal folds, creating an acoustic signal with pitch, loudness and voice quality. The excitation signal is then changed by the shape of the vocal tract and placement of the vocal articulators, which can be seen as an impulse response changing the acoustic waveform. The discrete-time filter model for the production of speech is shown in Figure 1.4. To more accurately model phoneme structures, the effect of the vocal tract on the speech signal must be separated from the contribution of the excitation signal.

To filter out the excitation signal, the speech signal $s(n)$ must be converted to a domain where it contains a linear combination of the excitation signal $e(n)$ and the vocal tract impulse response $\theta(n)$, or

$$s'(n) = e'(n) + \theta'(n) \quad (2.9)$$

where $e(n)$ and $\theta(n)$ are easily distinguished in the frequency domain. In the time domain the speech signal is considered to be the convolution of the excitation signal and the vocal tract impulse response.

$$s(n) = e(n) * \theta(n) \quad (2.10)$$

Taking the discrete Fourier transform of the speech signal, the convolution in the time-domain becomes multiplication in the frequency domain. Therefore

$$S(\omega) = E(\omega)\Theta(\omega) \quad (2.11)$$

Taking the logarithm of the magnitude spectrum $S(\omega)$, the result is a linear combination of the effects of the excitation signal and the vocal tract impulse response.

$$\log |S(\omega)| = \log |E(\omega)| + \log |\Theta(\omega)| \quad (2.12)$$

The log-power domain simplifies the separation of the effects of different contributors to the signal because they become additive constants that can easily be removed. Taking the Fourier transform of the logarithm of the magnitude spectrum results in a cepstrum, also called a spectrum of a spectrum. The speech signal is hereby converted to the cepstral domain. By using only the magnitude of the initial spectrum, the original signal cannot be recovered, resulting in a “real” cepstrum, but discarding the phase structure of the initial spectrum is acceptable for speech signal analysis because the human perception

system is unable to detect phase-structure in a speech signal. Because the excitation signal varies much faster in time due to pitch frequencies than the changes of the vocal tract response, the cepstrum of the vocal tract response is easily isolated by extracting only the lower portion of the cepstrum of the speech signal. In the cepstral domain this process is referred to as “liftering”.

When working with speech signals the spectrum is usually transformed using the Mel scale, a non-linear frequency scale. The Mel scale (based on the word “melody”) is a perceptual scale of equal pitch increments relating non-linearly to the actual frequencies. The Mel scale of frequency therefore more closely resembles the human auditory response than the linearly-spaced frequency bands used in normal cepstrum analysis. The Mel scale is linear in the low frequency range (below 1000Hz) and logarithmic above 1000Hz. The relationship between the Hertz scale and the Mel scale can be defined as

$$m = \frac{1000}{\log_{10} 2} \left(\log_{10} \left(1 + \frac{f}{1000} \right) \right) \quad (2.13)$$

where m is the Mel frequency and f is the normal Hertz frequency. This relationship is shown in Figure 2.2.

A discrete cosine transform (DCT) can be used on the spectrum instead of a discrete Fourier transform. The end result is the Mel frequency cepstrum (MFC).

A common feature extraction technique is to take successive overlapping windowed portions of the speech signal, typically 25 ms long, and finding the Mel frequency Cepstral Coefficients (MFCCs) of these sets. MFCCs are sensitive to the presence of additive noise in the speech signal, which increases the importance of noise-reduction in signal pre-processing. There are also variations on the basic MFC algorithm that can increase robustness specifically for use in speech recognition [81].

Feature Postprocessing

After calculating the parameters for the D -dimensional feature vectors from overlapping sets of frames, these features are processed to improve robustness, computational efficiency and accuracy for speech recognition applications. Consider all the feature vectors as datapoints in a D -dimensional space. Each axis represents the information associated with one of the parameters of the feature space. By looking at all the datapoints together as a cluster, the feature vectors can be normalised and transformed to improve recognition accuracy.

Unity variance normalisation is used to normalise the feature vectors by scaling each dimension so that the standard deviation of the datapoint density is unity in that direc-

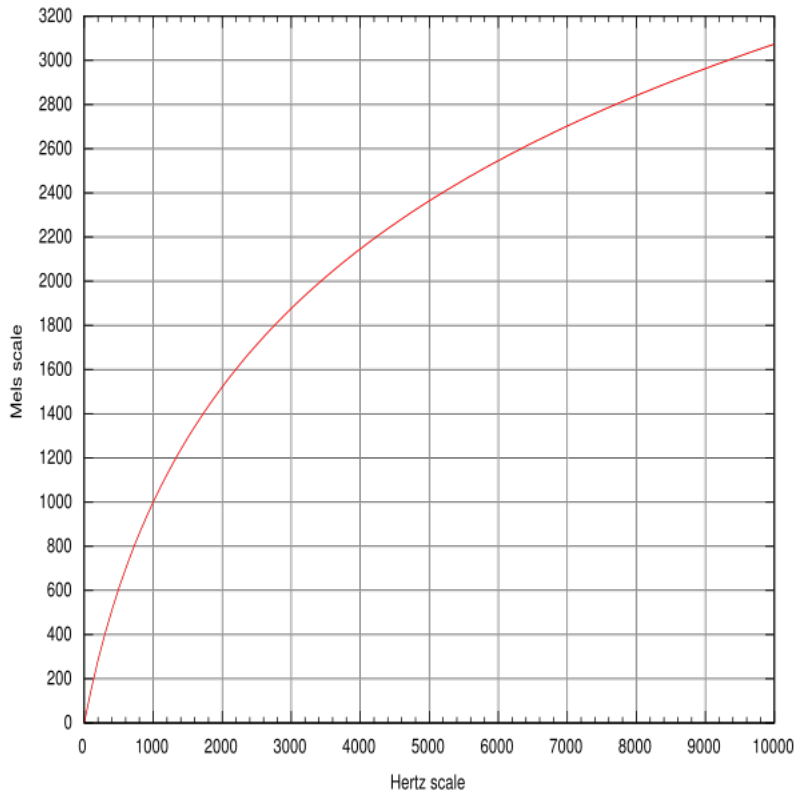


Figure 2.2: *Relationship between the Mel frequency scale and the Hertz frequency scale*

tion. This normalisation step will prevent numerical instability in ill-conditioned covariance matrices used during statistical modelling.

Cepstral mean removal is a technique used to remove the effect of additive noise in the cepstral domain by subtracting the mean of the cepstral feature values on each dimension from the feature vectors. This leads to increased robustness in the presence of channel noise and better comparability between feature vectors extracted from signals originating in different recording conditions. This technique requires the availability of all speech data prior to processing, but for applications where processing is synchronous with recording a running mean can be computed from the n last frames.

Incorporation of context is used to augment the current feature vector with its neighbouring feature vectors. To simplify calculations, speech recognition applications often use the assumption that successive feature vectors are conditionally independent from each other. This is a reasonable assumption, given the piecewise linear characteristics

of a speech signal broken down into sufficiently small segments, but is not absolutely accurate because of the smooth shape of the spectrum imposed by the vocal tract and the consequent relationships between successive spectral estimates. Incorporating neighbouring features partially overcomes the limitations set by the conditional independence assumption. Care needs to be taken at the edge of the signal where the adjacent vectors do not exist.

Dimension reduction is used to reduce the dimension of the feature vectors to increase computational efficiency and minimise the amount of training data needed to properly estimate all the parameters in the feature vector. After appending the first and second derivatives of the features (Δs and $\Delta\Delta s$) the dimensionality of the feature vectors is tripled. Depending on the data, some dimensions contain more information that can be used to discriminate between the classes of phonemes to be recognised than others. *Linear Discriminant Analysis* (LDA) is used to reduce the dimension of the feature data in such a way that loss of information required for discriminating between different phoneme classes are minimised. The feature vectors undergo linear transformations that essentially change the perspective on the data, to maximise the distance between the different phoneme classes. The features are then rotated so that the axis with maximum class information corresponds to one of the basis vectors in a new uncorrelated feature space. The amount of class information contained in a dimension of the feature vectors are determined with the use of eigenvalues with the largest eigenvalue corresponding to the dimension containing the most class information. After linear transformation, dimensions containing little class information can safely be discarded with minimal loss in recognition accuracy.

2.3.4 Acoustic Modelling

Acoustic modelling is the process of creating a statistical representation of a set of sub-speech units, typically based on phonemes, to be used in a speech recognition engine. Feature vectors obtained from a training data set are labelled according to the subword unit they belong to through the use of manual transcriptions. Manual transcriptions often contain only word sequences without any fixed boundaries between them, because obtaining manual transcriptions that accurately contain boundaries on subword level are both expensive and time-consuming. In the absence of accurate boundaries, bootstrapping techniques containing multiple steps of phoneme boundary re-estimation are needed to approximate the phoneme labels connected to a set of feature vectors. Once all the feature vectors belonging to a specific class are accumulated, a statistical model can be derived to represent that specific class. Many underlying principles of acoustic modelling are

also valid for acoustic decoding, when the trained statistical models are used to evaluate new speech data. Acoustic decoding will therefore not be considered separately in this high-level discussion.

Statistical Models

Acoustic models are very sensitive to the quality and type of the speech signal used as input. Evaluating speech data that differ from the training data in recording circumstances and type of speech corpora results in significantly lower recognition accuracies. The training data set must therefore be collected and processed in such a way as to approximate as closely as possible the conditions of the data that will be evaluated later.

A direct approach to statistical modelling would be to collect all data for a specific class and create a single probability distribution model that best fits the data, ideally using a distribution function that closely approximates natural data, for example the Gaussian distribution model. Assuming that enough datapoints are provided, the model would be robust in the presence of outliers. However, the distribution in the D -dimensional feature space can have a complex structure and can perhaps be better modelled using a set of weighted Gaussian distribution models. Gaussian Mixture Models (GMMs) are used extensively in speech recognition applications with good results.

One of the most popular statistical techniques used to create acoustic models for speech recognition is an evolution of GMM theory – the Hidden Markov Model (HMM). The use of HMMs are wide-spread because of their good performance and efficient estimation of model parameters based on well-known mathematical techniques. An HMM is a type of finite-state machine, which models the quasi-stationary nature of the speech signal on the states, each presented by a probability density function. The temporal structure of speech data are modelled in the transitions between these states. Hidden Markov models are explained in detail in Chapter 3.

Choice of Subword Units

The most widely used elementary acoustic units in continuous-speech recognition models are phoneme-based. The phonemes in a language are the smallest units of speech that represent unique sounds with no relevance to their positions in a word or phrase. Compared to larger units (syllables, demisyllables or words), phoneme models reduce the number of classes and consequently the number of parameters that need to be estimated, resulting in better computational efficiency and better use of limited training data. Additionally, using statistical models for whole words is not feasible for large vocabulary systems.

Phonemes can be modelled in isolation, creating monophone models, or modelled in

context with neighbouring phonemes, creating context-dependent models such as biphones and triphones. Phoneme models enable cross-word modelling, simplify porting for use in new vocabularies and can be used to model out-of-vocabulary words by implementing back-off mechanisms. In contrast with phoneme modelling, diphone modelling focuses on the transitions between phonemes rather than the phonemes themselves. Chapter 4 contains detailed comparisons between the different subword units used for speech recognition, focusing on the strengths and weaknesses of transition models and context-dependent models.

2.3.5 Lexical Modelling

Lexical modelling is concerned with the relationship between the acoustic-level representation of the signal and the word sequence output by the speech recogniser for a given language. Lexical modelling for a pre-defined vocabulary is described with a lexicon – a list of vocabulary items stated in terms of the basic acoustic units used by the recogniser.

The subword units used to describe each word in the lexicon correspond to the models generated during acoustic modelling. Pronunciation lexicons are usually created manually, but several approaches to automatic learning of lexical contexts have been investigated [75].

One of the most important aspects of lexical modelling is the lexical coverage. Words that are not defined in the lexicon are called out-of-vocabulary (OOV) words and without special handling can cause a significant drop in recognition accuracy. It is therefore beneficial to word recognition applications to maximise the coverage of the vocabulary, which is not a trivial task for large-vocabulary continuous-speech recognition, which handles vocabularies that can contain around 20 000 words.

Many words have alternative pronunciations for which additional entries are needed in the lexicon. An example is homographs (words that are spelled the same but pronounced differently) like *record* and *produce* where the pronunciation depends on the context of the word in the sentence structure. Other factors include differences in dialect and accents and shortened words due to the effects of co-articulation and deletion (*interest* versus *int'rest*).

2.3.6 Language Modelling

Language modelling is used to estimate the likelihood of word sequences generated through lower levels of acoustic and lexical modelling, to constrain the output to a word sequence that makes sense in the given language. The introduction of grammatical constraints for

use with small vocabularies can be done manually, but complex language models used in large vocabularies need to be modelled stochastically. The most popular statistical language models are n -gram models, which estimate the frequency of occurrence for a sequence of n words. In these models the probability of a word string is reduced to finding the probability of consecutive sets of n sequential words, reducing the word history at each word to the preceding $n - 1$ words.

For large vocabularies (> 10000) and higher-order n -gram modelling ($n > 3$) an extremely large amount of training data is required to model all possible n -sized sequences. Back-off mechanisms can be used to smooth estimates of rare n -grams by using lower order n -grams which should have more training data available. For example in bigram language modelling ($n = 2$), the number of occurrences of each word pair (w_{i-1}, w_i) , normalised by the number of occurrences of the preceding word w_{i-1} , is counted to provide a measure of the probability of word w_i following word w_{i-1} . 0-gram modelling equates to the situation where any word in the vocabulary can be followed by any other word in the vocabulary, with a probability related to its own frequency in the training data.

Language models for spontaneous speech have many problems that need to be addressed, such as the handling of sounds that are not speech (laughter, coughing), compound words, acronyms, filled pauses or speech disfluencies (*uh*, *um* and *erm*). Important steps when training a language model is garbage bracketing, which removes all material not suited for sentence-based language modelling, and the handling of errors in the training text, for example spelling and typing errors.

The variability of human speech complicates the process of linguistic modelling, making large-vocabulary speech recognition truly a challenge.

2.4 Summary

This chapter provides the theoretical background of statistical speech recognition as used in this thesis, starting with a general overview of the recognition problem and its numerous challenges. With many differences between speech recognition applications in terms of variables such as the type of speech corpora used, vocabulary size, speaker-independence or -dependence and signal recording conditions, direct comparisons are difficult. Through the years speech recognition technology evolved from the recognition of microphone-recorded, speaker-specific, isolated utterances of digits to the recognition of telephone-recorded, speaker-independent sentences produced from fluent continuous-speech.

One of the areas of speech recognition where significant progress has been made is acoustic modelling, where a set of subword units are modelled and concatenated to create

word models. Transitional models, such as diphones, have been investigated through different methodologies, including diphone templates, transitional subspace models, artificial neural networks and hidden Markov models. A brief literature study of speech recognition research related to diphones has shown that a diphone-based system can outperform a basic phoneme-based system, as well as recognition systems based on context-dependent models such as biphones and triphones.

The rest of the chapter is dedicated to a discussion of the different components of a speech recognition system. The main components are:

- **Digital signal processing** of the acoustic signal, including signal preprocessing, feature extraction and feature postprocessing. The signal preprocessing front-end is used to obtain the valuable information pertaining to the articulatory configuration and resulting sounds, while suppressing contributions made by differences in speaker style, prosody and recording conditions. Feature extraction digitises the signal and extracts numerical representations of spectral characteristics of short segments of the speech signal by first transforming the signal to the cepstral domain, creating Mel frequency Cepstral Coefficients (MFCCs). Feature postprocessing consists of unity variance normalisation, cepstral mean removal, incorporation of neighbouring feature frames and dimension reduction. The final result is a sequence of numerical feature vectors, each representing about 10 milliseconds of speech, that can be used in the acoustic modelling process.
- **Acoustic modelling**, used to create statistical representations of each subword class in the recognition system. The type and configuration of the statistical model is an important consideration, as well as the choice of subword unit to use for acoustic modelling.
- **Lexical modelling**, used to define the morphological relationship between the subword-level acoustic models to create the words in a given language.
- **Language modelling**, used to define the syntactic relationship between the word models to simulate the rules governing sentences in spoken language.

The research done in this thesis only pertains to acoustic modelling, which is discussed further in the following chapters.

Chapter 3

Hidden Markov Model Theory

A Hidden Markov Model (HMM) is a statistical model structured as a finite state machine where the current state is not observable. Instead the variables influenced by the state are observable.

An HMM is a popular choice for modelling speech and other temporal signals because of its ability to simulate the dynamic nature of signals whose structure changes with time in a semi-predictable manner. Hidden Markov Models also have a rich mathematical structure, which provides a solid base for statistical modelling and is used for a variety of statistical pattern recognition applications, including DNA research, cryptanalysis (code breaking), image recognition and machine learning. The theory was first described in a series of statistical papers by Leonard E. Baum and other authors in the latter half of the 1960s [5, 6, 7]. Since then the use of HMMs for speech modelling has become commonplace. Many algorithms were developed to utilise HMM theory more efficiently, making HMM research a valuable research topic. A good resource for fundamental HMM theory can be found in [64].

A state transition machine model is at the core of HMM theory. A finite set of states represents the valid conditions a system can be in, with defined transitions between these states. The state transition machine, also called a graph or automaton, is a basic tool used in computer science and information theory. It has the ability to both model stationary characteristics within states and transitional characteristics between states.

In many statistical modelling applications it's easier to assume that the data samples, or feature vectors, are independent and identically distributed (IID assumption). However, for speech recognition a single distribution function used to model all feature vectors obtained from a speech segment is far too general and fails to model the dynamic changes that characterise human speech. At the other end of the complexity scale we can try to model each feature vector with its own probability density. This would not aid us in generating a general model for a segment of speech, but rather result in a highly specialised

one. The HMM approach to the modelling of speech data is a compromise between these two extremes. Segments of speech are assumed to have subsections that can be modelled using the IID assumption if the subsections are sufficiently small.

3.1 Definition of a Hidden Markov Model

3.1.1 Markov Chains

A first order Markov chain is defined as a stochastic process with the Markov property, implying that future states are only influenced by the current state and not by past states. It is therefore a parametric random process where future states are reached through a probabilistic process rather than a deterministic one.

Let $X_1, X_2, X_3, \dots, X_n$ be a sequence of random variables all with the same sample space \mathcal{X} . The probability of the sequence is expressed with Bayes' formula as

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | X_1, X_2, \dots, X_{i-1}). \quad (3.1)$$

The probability of a random variable is dependent on the probabilities of preceding random variables in the sequence. For first order Markov chains, however, the probability of a random variable is only dependent on the previous random variable and not on the complete sequence. Therefore,

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | X_{i-1}). \quad (3.2)$$

The Markov chain is completely specified by the transition functions – the set of probabilities for moving from one variable to the next.

$$P(X_i = x' | X_{i-1} = x) = p(x' | x) \quad \text{for all } x, x' \in \mathcal{X} \quad (3.3)$$

The transition function $p(x' | x)$ must satisfy the usual conditions for probabilities: All transitions extending from a specific variable must sum to 1

$$\sum_{x' \in \mathcal{X}} p(x' | x) = 1 \quad (3.4)$$

and all transition probabilities are positive real values.

$$p(x' | x) \geq 0 \quad x' \in \mathcal{X} \quad (3.5)$$

Markov chains are not limited to processes with a one-step memory. They are in fact capable of modelling processes with arbitrary complexity. Higher order Markov chains can be built using a memory length larger than 1 by defining new random variables equal

to the k -length sequence of random variables preceding the current random variable. By extending the sample set to include these possible sequences, the mathematical theory holds.

3.1.2 Hidden Markov Models

A Hidden Markov model is a statistical model where the underlying system is a Markov chain with hidden internal parameters. The system generates observable data which can be used to determine the underlying parameters. Once the internal parameters are defined, the model is used to determine the similarity of a previously unseen pattern to the model, effectively classifying the pattern.

The HMM can be seen as a state transition machine with a finite set of N states, each one corresponding to a stochastic time-invariant random process in a Markov chain. The states are defined as a set Q where

$$\mathbf{Q} = \{q_1, q_2, \dots, q_N\}. \quad (3.6)$$

The observable data is a sequence of observation symbols (feature vectors) of length T .

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\} \quad (3.7)$$

The hidden state sequence corresponding to the sequence of observations is defined as set S where

$$\mathbf{S} = \{s_1, s_2, \dots, s_T\} \quad \text{for observation sequence of length } T. \quad (3.8)$$

The transition probabilities $p(q_i|q_j)$ between these states are defined as the conditional probability of the model being in state q_j at time t , given that it was in state q_i at time $t-1$. The transition probabilities of the HMM models the temporal structure of the data. These probabilities are grouped together in a state probability transition matrix A where the entry $a_{ij} = p(q_i|q_j)$

$$\mathbf{A} = [a_{ij}], \quad a_{ij} = P(s_t = q_j | s_{t-1} = q_i), \quad i, j = 1, 2, \dots, N \quad (3.9)$$

Each state in the model has an associated output probability distribution that is defined as the conditional probability of the feature vector \mathbf{x}_t , given that the system is currently in the specific state. The conditional output probability density models the quasi-stationary character of the input sequence, allowing similar feature vectors to be clustered together within the temporal structure of the data. The output probability distributions are defined as a set B where

$$\mathbf{B} = \{b_j(\mathbf{x}_t)\}, \quad b_j(\mathbf{x}_t) = f(\mathbf{x}_t | s_t = q_j), \quad j = 1, 2, \dots, N. \quad (3.10)$$

The HMM must have well-defined starting and ending states. The set of probabilities for initially starting in a specific state is defined as π where

$$\pi = \{\pi_i\}, \quad \pi_i = P(q_0 = s_i). \quad (3.11)$$

An HMM is completely specified by its three sets of probabilities. It is therefore represented as

$$\Phi = (\mathbf{A}, \mathbf{B}, \pi). \quad (3.12)$$

Two basic assumptions are used when working with hidden Markov models - the Markov assumption and the output independence assumption.

The Markov assumption stems from the Markov property of Markov chains. It states that the conditional probability of being in state s_t at time t is only dependent on the R previous states where R is the order of the Markov model. For a first-order HMM,

$$P(s_t | s_1^{t-1}) = P(s_t | s_{t-1}). \quad (3.13)$$

For an HMM of order R ,

$$P(s_t | s_1^{t-1}) = P(s_t | s_{t-R}^{t-1}). \quad (3.14)$$

The output independence assumption states that the conditional density at time t of a vector \mathbf{x}_t given the state sequence s_1^t is only dependent on the current state. Mathematically

$$f(\mathbf{x}_t | \mathbf{x}_1^{t-1}, s_1^t) = f(\mathbf{x}_t | s_t). \quad (3.15)$$

If we know the probable state sequence and hence the state and density each observation vector belongs to, the total probability of the sequence of vectors is reduced to the product of the individual probabilities.

A typical HMM structure used for speech recognition is shown in Figure 3.1. The model shown has three internal states (q_1, q_2 and q_3), called emitting states, and two null states denoting the start and end of the model. Null states do not have associated output probability densities and are used to define transitions in the HMM structure that are not accompanied by a step in time to the next observation sequence. The entries of the state transition matrix \mathbf{A} are shown as transition probabilities $a_{11}, a_{12}, \dots, a_{33}$ where the transitions not shown (such as a_{21}) are assumed to have a probability of 0. Each emitting state has its own output probability density function expressed by the values of $\mathbf{b}_1, \mathbf{b}_2$ and \mathbf{b}_3 . State q_1 is defined as the only valid starting node with starting probability $\pi_1 = 1$. The probability of starting with any other node is 0. State q_3 is defined as the only valid ending node, with a link to the ending null state.

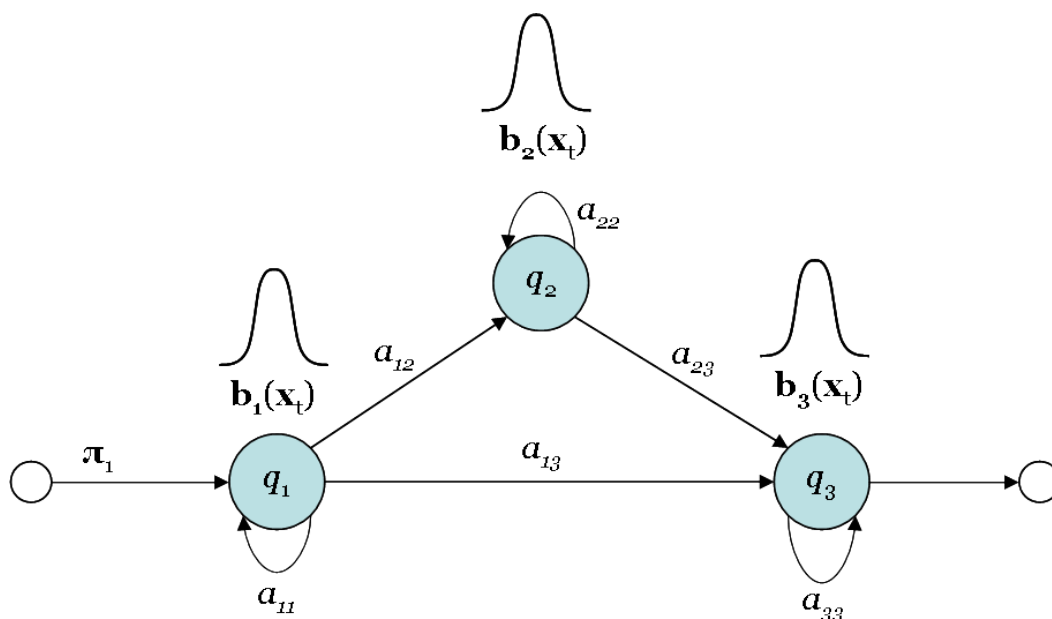


Figure 3.1: 3 State left-to-right Hidden Markov Model

3.2 Algorithms Used with Hidden Markov Models

In practical applications Hidden Markov Models have three fundamental problems:

1. **The evaluation problem:** Given the observation sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ and the model parameters, how do we compute the total probability of the observation sequence $P(\mathbf{X}|\Phi)$? With known model parameters and an estimated state sequence (hidden information), the similarity of the observation sequence to the model can be calculated. Therefore it can be seen as a classification problem.
2. **The decoding problem:** Given the observation sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ and the model parameters, how do we find the optimal state sequence $\mathbf{S} = (s_1, s_2, \dots, s_T)$? This is a problem of finding the hidden information in the HMM if the model parameters are known.
3. **The learning problem:** Given the observation sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ and the model parameters, how do we optimise the model by adjusting the parameters to maximise $P(\mathbf{X}|\Phi)$?

3.2.1 The Evaluation Problem

The probability of a given observation sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ is equal to the probability of the observation sequence given the state sequence $\mathbf{S} = (s_1, s_2, \dots, s_T)$ for the observations multiplied by the probability that the state sequence was generated.

$$P(\mathbf{X}, \mathbf{S} | \Phi) = P(\mathbf{X} | \mathbf{S}, \Phi) P(\mathbf{S} | \Phi) \quad (3.16)$$

Using the output independence assumption,

$$P(\mathbf{x}_t | \mathbf{x}_1^{t-1}, s_1^t) = P(\mathbf{x}_t | s_t). \quad (3.17)$$

the probability of the observation sequence given the state sequence is

$$\begin{aligned} P(\mathbf{X} | \mathbf{S}, \Phi) &= P(\mathbf{x}_1 | s_1) P(\mathbf{x}_2 | \mathbf{x}_1, s_1^2) P(\mathbf{x}_3 | \mathbf{x}_1^2, s_1^3) \cdots P(\mathbf{x}_T | \mathbf{x}_1^{T-1}, s_1^T) \\ &= \prod_{t=1}^T P(\mathbf{x}_t | \mathbf{x}_1^{t-1}, s_1^t) \\ &= \prod_{t=1}^T P(\mathbf{x}_t | s_t) \\ &= \prod_{t=1}^T b_{s_t}(\mathbf{x}_t) \\ &= b_{s_1}(\mathbf{x}_1) b_{s_2}(\mathbf{x}_2) \cdots b_{s_T}(\mathbf{x}_T) \end{aligned} \quad (3.18)$$

Using the first order Markov assumption,

$$P(s_t | s_1^{t-1}) = P(s_t | s_{t-1}), \quad (3.19)$$

the probability of the state sequence is

$$\begin{aligned} P(\mathbf{S} | \Phi) &= P(s_1) P(s_2 | s_1) P(s_3 | s_1^2) \cdots P(s_T | s_1^{T-1}) \\ &= P(s_1) \prod_{t=2}^T P(s_t | s_1^{t-1}) \\ &= P(s_1) \prod_{t=2}^T P(s_t | s_{t-1}) \\ &= \pi_{s_1} a_{s_1 s_2} a_{s_2 s_3} \cdots a_{s_{T-1} s_T} \end{aligned} \quad (3.20)$$

Therefore substituting Equations 3.18 and 3.20 into 3.16, the total probability of an observation sequence is the sum over all possible state sequences

$$\begin{aligned} P(\mathbf{X} | \Phi) &= \sum_{\text{all } \mathbf{S}} P(\mathbf{X} | \mathbf{S}, \Phi) P(\mathbf{S} | \Phi) \\ &= \sum_{\text{all } \mathbf{S}} \pi_{s_1} b_{s_1}(\mathbf{x}_1) a_{s_1 s_2} b_{s_2}(\mathbf{x}_2) a_{s_2 s_3} \cdots a_{s_{T-1} s_T} b_{s_T}(\mathbf{x}_T) \end{aligned}$$

The Forward Algorithm

The complexity of these calculations are very high and computationally expensive, especially with large models or long observation sequences. There are N^T possible state sequences resulting in exponential computational complexity. However, the complexity of the problem can be reduced by using the time invariance of the probabilities. This process is called the forward algorithm.

We first define a partial probability $\alpha_t(i)$ as the probability of reaching the intermediate state i at time t .

$$\alpha_t(j) = P(\mathbf{x}_1^t, s_t = j, |\Phi) \quad (3.21)$$

The partial probability of a state at time t can be recursively defined in terms of the partial probabilities of all states at time $t - 1$.

$$\alpha_t(j) = P(\mathbf{x}_t | s_t = j, \Phi) P(\text{all paths to state } i \text{ at time } t) \quad (3.22)$$

The first term in the product is known through the output probability distribution matrix. To calculate the probability of reaching the intermediate state at time t , we sum the partial probabilities of all the states at time $t - 1$ multiplied by the respective transition probabilities in the transition matrix. Formulation of the partial probability of state j at time t in terms of the partial probabilities at time $t - 1$ is as follows:

$$\begin{aligned} \alpha_t(j) &= P(\mathbf{x}_1^t, s_t = j, |\Phi) \\ &= \sum_{s_1^{t-1}} P(\mathbf{x}_1^t, s_1^{t-1}, s_t = j | \Phi) \\ &= \sum_{s_1^{t-1}} P(\mathbf{x}_1^t | s_1^{t-1}, s_t = j, \Phi) P(s_1^{t-1}, s_t = j | \Phi) \\ &= \sum_{s_1^{t-1}} P(\mathbf{x}_t | s_t = j, \Phi) P(\mathbf{x}_1^{t-1} | s_1^{t-1}, s_t = j, \Phi) P(s_1^{t-1}, s_t = j | \Phi) \\ &= b_j(\mathbf{x}_t) \sum_{s_1^{t-1}} P(\mathbf{x}_1^{t-1} | s_1^{t-1}, s_t = j, \Phi) P(s_t = j | s_1^{t-1}, \Phi) P(s_1^{t-1} | \Phi) \\ &= b_j(\mathbf{x}_t) \sum_i \sum_{s_1^{t-2}} \left[P(\mathbf{x}_1^{t-1} | s_1^{t-2}, s_{t-1} = i, s_t = j, \Phi) P(s_t = j | s_{t-1} = i, s_1^{t-2}, \Phi) \right. \\ &\quad \left. P(s_1^{t-2}, s_{t-1} = i | \Phi) \right] \\ &= b_j(\mathbf{x}_t) \sum_i a_{ij} \sum_{s_1^{t-2}} P(\mathbf{x}_1^{t-1} | s_1^{t-2}, s_{t-1} = i, s_t = j, \Phi) P(s_1^{t-2}, s_{t-1} = i | \Phi) \\ &= b_j(\mathbf{x}_t) \sum_i a_{ij} \sum_{s_1^{t-2}} P(\mathbf{x}_1^{t-1}, s_1^{t-2}, s_{t-1} = i | \Phi) \\ &= b_j(\mathbf{x}_t) \sum_i a_{ij} P(\mathbf{x}_1^{t-1}, s_{t-1} = i | \Phi) \\ &= b_j(\mathbf{x}_t) \sum_i a_{ij} \alpha_{t-1}(i) \end{aligned} \quad (3.23)$$

This process is illustrated in Figure 3.2.

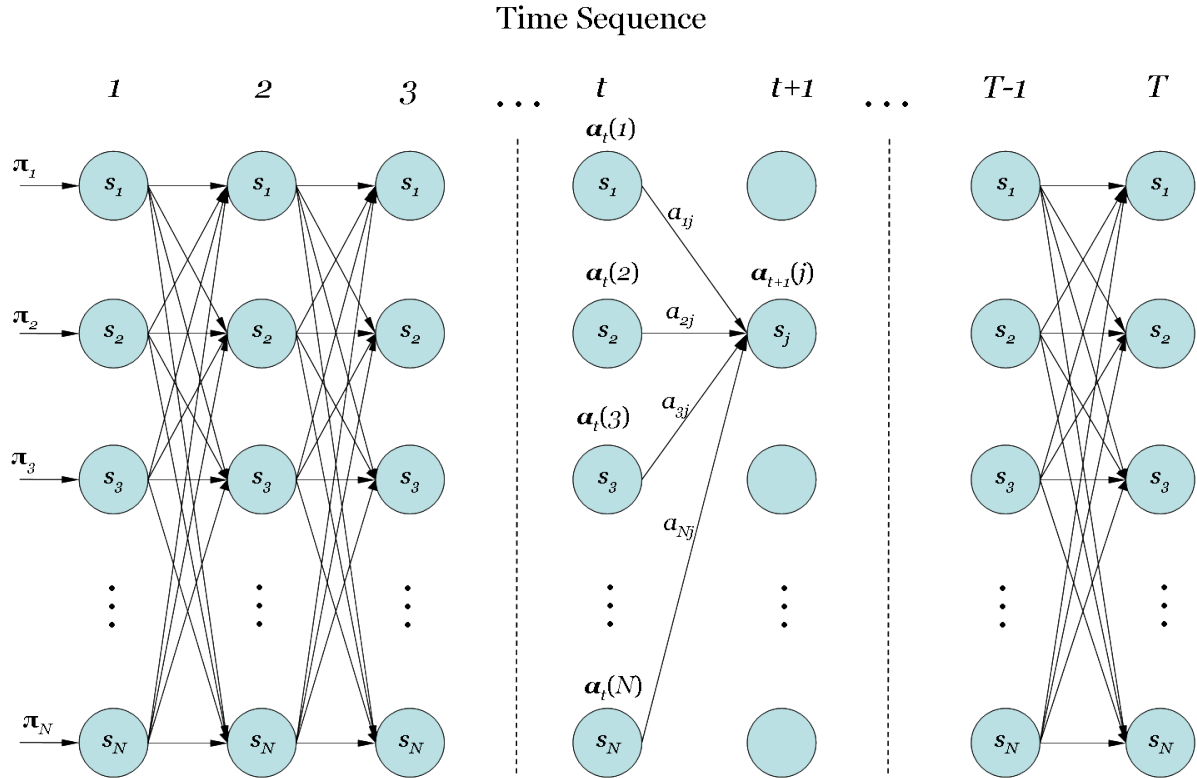


Figure 3.2: *The forward algorithm. The partial probability $\alpha_{t+1}(j)$ is recursively defined by multiplying the output probability of state s_j with the sum of the partial probabilities of the states leading to state s_j multiplied by their respective transition probabilities.*

Using partial probabilities recursively we can calculate the total probability of the observation sequence. The probability of generating the observation sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ given the model Φ is equal to the sum of all the partial probabilities at time T . Only partial probabilities that are linked to terminating states in the HMM at time T are considered valid. Therefore the total probability of the observation sequence is the sum of the partial probabilities at time T of all the terminating states.

$$P(\mathbf{X}|\Phi) = \sum_{i=1}^N \alpha_T(i) \quad (3.24)$$

The partial probability is recursively defined as

$$\alpha_{t+1}(j) = b_j(\mathbf{x}_{t+1}) \sum_{i=1}^N \alpha_t(i) * a_{ij}, \quad 1 \leq j \leq N; 1 \leq t \leq T \quad (3.25)$$

with starting conditions

$$\alpha_1(j) = b_j(\mathbf{x}_1)\pi_j, \quad 1 \leq j \leq N. \quad (3.26)$$

By visualising the process in a grid pattern as illustrated in Figure 3.2, the reduction in computational complexity can be seen. This grid pattern is a common way of visualising HMM algorithms and is called a *trellis*. The states of the HMM model and their transitions are shown in a time-synchronous fashion. For the forward algorithm the probabilities of being in each state at a time t is calculated before moving on to time $t + 1$. When all the columns have been evaluated, the sum of the probabilities in terminating states at time T is the probability that the HMM generated observation sequence \mathbf{X} .

The big-O notation is used as a theoretical measure of the complexity and execution of an algorithm, denoting the order of the function used to describe the amount of instructions needed to execute the algorithm. The computational complexity of the forward algorithm is $O(N^2T)$ for a fully connected model. To calculate a partial probability N multiplications are made, and there are $N * T$ partial probabilities to be calculated. This is a significant reduction from the complexity of $O(N^T)$ without the use of the forward algorithm.

3.2.2 The Decoding Problem

Finding the optimal state sequence, given a model Φ and an observation sequence \mathbf{X} is one of the fundamental operations done in HMM theory. There are several ways to define an “optimal” state sequence, leading to various possible solutions to the decoding problem. One possibility is to find the state sequence where the individual states are considered invariant, calculating the most likely state at each time interval. The problem with this approach is that not all models are fully connected, containing transitions with 0 probability. The sequence of states which are individually most likely may turn out to be an invalid state sequence. The most widely used optimality criterion is to consider the entire sequence as a whole. Similar to the forward algorithm described above, the Viterbi Algorithm is used to evaluate all possible paths through the HMM in order to find the most optimal state sequence. The optimal state sequence is the one with the highest probability of being taken while generating the observation sequence.

Noting that

$$P(\mathbf{S}|\mathbf{X}, \Phi) = \frac{P(\mathbf{S}, \mathbf{X}|\Phi)}{P(\mathbf{X}|\Phi)}, \quad (3.27)$$

finding the state sequence $\mathbf{S} = (s_1, s_2, \dots, s_T)$ that maximizes $P(\mathbf{S}, \mathbf{X}|\Phi)$ will also maximize the state sequence given the observation sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$, yielding the

optimal state sequence. Also mathematically described as

$$\begin{aligned} \mathbf{S} &= \arg \max_{s_1^T} P(s_1^T | \mathbf{x}_1^T, \Phi) \\ &= \arg \max_{s_1^T} \frac{P(s_1^T, \mathbf{x}_1^T | \Phi)}{p(\mathbf{x}_1^T)} \end{aligned} \quad (3.28)$$

The Viterbi Algorithm

The Viterbi algorithm is an algorithm used in dynamic programming for finding the most likely path, also called the Viterbi path, in a set of possible state sequences. The Viterbi algorithm relies on the Markov assumption as well as the assumption that the two sequences, in this case the observation sequence \mathbf{X} and the state sequence \mathbf{S} are aligned. For the case of the HMM we can assume that the two sequences have the same length T corresponding to the time-sequence.

Using the Markov assumption the Viterbi algorithm evaluates the sequences in a time synchronous fashion. The Viterbi algorithm is very efficient and there are modifications that reduce computational load, but the memory required to store the path history tend to remain constant.

Through modification of the forward algorithm, we can define a partial best-path probability $\delta_t(i)$ as the probability of the most likely state sequence up to time t , which generated the partial observation sequence \mathbf{x}_1^t and ends in state i .

$$\delta_t(i) = \max_{s_1^{t-1}} P(\mathbf{x}_1^t, s_1^{t-1}, s_t = i, | \Phi) \quad (3.29)$$

Or alternatively

$$\delta_t(i) = P(\mathbf{x}_t | s_t = i, \Phi) * \max_{s_1^{t-1}} P(\text{all paths leading to state } i \text{ at time } t). \quad (3.30)$$

The first term in the product is known through the output probability distribution matrix. To calculate the probability of the best path leading to intermediate state i at time t , we find the maximum partial best-path probabilities of all the states at time $t - 1$ multiplied by the respective transition probabilities in the transition matrix.

Using partial best-path probabilities recursively we can calculate the best path through the trellis for the entire observation sequence. In order to keep track of the best path, a matrix $B_t(i)$ is used to store, for each state i at time t , the number of the state with maximum probability of being the previous state in the partial best-path sequence. The last state of the optimal state sequence s'_T can be found by noting the state at time T with maximum partial best-path probability. From this last state, backtracking is used to step through the $B_t(i)$ matrix to find the most likely previous state for each

time interval up to the beginning. Only partial best-path probabilities that are linked to terminating states in the HMM at time T are considered valid. The total probability of the observation sequence P'_T , given the optimal state sequence S' is the maximum partial best-path probability at time T of all the terminating states.

$$P'_T = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (3.31)$$

$$s'_T = \arg \max_{1 \leq i \leq N} B_T(i) \quad (3.32)$$

The partial best-path probability is recursively defined as

$$\delta_{t+1}(j) = b_j(\mathbf{x}_{t+1}) \max_{1 \leq i \leq N} [\delta_t(i) * a_{ij}], \quad 1 \leq j \leq N; 1 \leq t \leq T \quad (3.33)$$

with starting conditions

$$\delta_1(j) = b_j(\mathbf{x}_1) \pi_j, \quad 1 \leq j \leq N \quad (3.34)$$

and population of the backtracking matrix $B_t(i)$ is done with

$$B_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) * a_{ij}]. \quad (3.35)$$

The complexity of the viterbi algorithm is $O(N^2T)$ for a fully connected model. To calculate a partial probability N multiplications are made, and there are $N * T$ partial probabilities to be calculated.

3.2.3 The Learning Problem

The most challenging problem of hidden Markov models is the fine-tuning of the model parameters to maximise a given observation sequence. An Expectation-Maximisation (EM) algorithm is used to iteratively improve the parameters of the model in order to locally maximise the likelihood of the observation sequence, or $P(\mathbf{X}|\Phi)$. One such EM-algorithm, the Baum-Welch algorithm, is commonly used to solve the learning problem.

Expectation-Maximisation Algorithm

Expectation-Maximisation (EM) algorithms are used to obtain maximum likelihood estimates of parameters where the observed information is considered as “incomplete” in the sense that it inherits its characteristics by way of a mapping from a larger complete model [25]. Typically, maximum likelihood estimation (MLE) cannot be done on the incomplete data, but is well defined on the larger complete model. In the HMM the observed data are directly influenced by the hidden information in the HMM, the state sequence. Therefore in order to find the maximum likelihood estimation of the model parameters, we need to find the hidden information first.

The EM algorithm alternates between two steps. The Expectation step, or E-step, calculates an estimation of the likelihood of the observed data, given an approximate model. The Maximisation step, or M-step, calculates revised parameters for the model based on maximising the expected likelihood calculated in the E-step. Starting with an initial set of parameters, the EM algorithm iterates over the two steps until convergence occurs according to predefined convergence criteria, usually associated with a minimal change in the calculated likelihood of the observed data. The exact measure of minimal change should be chosen to make the algorithm efficiently fast, while still remaining robust. A very important consideration when working with an EM algorithm is what to use as the initial set of parameters. Because the EM algorithm aims to find a local maximum likelihood, different starting points may lead to different convergence points, or local maxima.

The Baum-Welch Algorithm

The Baum-Welch algorithm is an example of an EM algorithm applied to hidden Markov models. The parameters of the HMM $\Phi = (\mathbf{A}, \mathbf{B}, \pi)$ can be iteratively refined by maximising the likelihood $P(\mathbf{x}|\Phi)$ in each iteration. When using the Baum-Welch algorithm it can be shown that

$$P(\mathbf{X}|\hat{\Phi}) \geq P(\mathbf{X}|\Phi) \tag{3.36}$$

where $\hat{\Phi}$ is the new set of HMM parameters after one iteration of the EM algorithm.

During the E-step of the Baum-Welch algorithm, an optimal state sequence is found that fits the observation sequence onto the model parameters. Using this information, maximum likelihood estimation is done for the three model parameters \mathbf{A} , \mathbf{B} and π individually. To calculate the starting probabilities the number of states being used as starting states are counted and factored into the existing starting state probabilities. The parameters of the densities attached to each state are calculated using all observation vectors that belong to that state according to the given state sequence. The same procedure is followed to calculate the transition probabilities between states.

Up to now we have considered using only one observation sequence as training data. The algorithm is easily generalised to use multiple observation sequences, assuming the observation sequences are statistically independent of each other. To train an HMM from M observation sequences is equivalent to finding the HMM parameters that maximises the joint probability

$$\prod_{m=1}^M P(\mathbf{X}_m|\Phi). \tag{3.37}$$

The Baum-Welch algorithm is performed on all the observation sequences to identify their hidden state sequence information. The information gathered from the E-step of all the sequences is then used to calculate the new parameter values $\hat{\Phi}$ before normalising the parameters to sum to one. Without normalising the model parameters, the partial forward probabilities used during the E-step lose precision after a large number of iterations. To prevent this loss from seriously affecting the estimates of the HMM parameters, normalisation during each iteration is essential.

3.3 Hidden Markov Models Used in Speech Recognition

In an automatic speech recognition system the HMM modelling stage is preceded by the preprocessing stage where the speech signal is analyzed and feature extraction is done. The input into the HMM is a time sequence of feature vectors as described in the previous chapter.

3.3.1 HMM Topology

All the algorithms discussed in the previous section rely on some knowledge of the structure of the HMM. The topology of a speech recognition HMM is representative of the way the HMM models the temporal characteristics of the speech signal as expressed in the feature vectors. It is directly influenced by its practical application where knowledge of the input data is essential to ensure an effective and robust HMM model. The topology of an HMM is reflected in the number of states chosen and the characteristics of the transition matrix A , which determines the transitions between states, including Null transitions that have a zero probability of occurring.

In speech recognition there are two HMM topologies that are used frequently: the fully-connected model and the left-to-right model. These elementary topologies were first used in the 1970s in speech applications and are still the preferred topologies today. The problem of dynamically finding optimal HMM structures is being addressed by researchers and partially solved using genetic algorithms or heuristic evaluation of complexity to prune an optimal HMM from a larger general structure. By allowing the data to reveal its own dynamic structure without external assumptions concerning the number of states or patterns of transitions, a slight improvement in recognition accuracy is achieved. Algorithms used to dynamically find optimal HMM topologies fail to provide significantly improved results despite the substantial increase in complexity [45], proving

the worth of the elementary HMM topologies used for over 30 years.

Fully Connected HMM

In a fully connected HMM model any state can be reached from any other state in a single time-step. Additionally, all states are both valid starting and valid terminating states. The fully connected model is non-restricting, with the only external knowledge applied to the structure being the number of states used. It yields the maximum number of possible paths through the HMM to generate a given observation sequence and is consequently the most resource intensive. All other first order HMM topologies are special instances of the fully connected model where some transition probabilities are set to zero.

A fully connected HMM topology is shown in Figure 3.3.

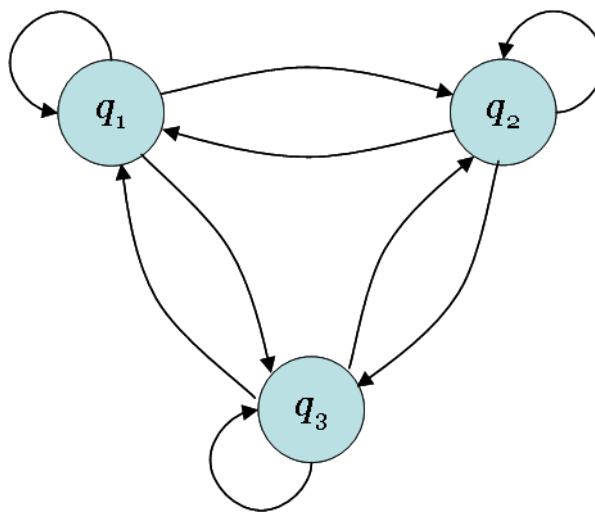


Figure 3.3: *3 State fully connected Hidden Markov Model*

These HMMs have a large number of parameters to be estimated, leading to a more complex likelihood function and an increase in the number of ill-suited local maxima. The freedom bestowed on the fully connected model to estimate its large number of parameters results in a poorly estimated model when the amount of available training data is limited. One way to avoid this situation is to use prior knowledge of the input data to design a better suited topology with fewer parameters. There are situations where using the fully connected topology cannot be avoided. In speech recognition applications it is commonly used for non-speech segments in the signal, such as silence, coughing, laughter or other external noises. These sounds do not have a pre-defined structure and are therefore best modelled with a general topology.

Left-to-Right HMM

The left-to-right HMM model, also called a Bakis model, is a very good topology to use for speech patterns, because it models input data that changes with time in a progressive manner. If a speech signal is broken down into small segments, as is the case in phoneme recognition, the phoneme segments themselves have a definite temporal structure. The left-to-right topology has the characteristic that the state index of the observation vectors stays the same or increases as the time index increases, making skipping back to previous states impossible. Consequently the transition probabilities from a higher state number to a lower state number are set to zero and the only valid starting state is the first state of the HMM.

$$a_{ij} = 0 \quad j < i \quad (3.38)$$

The state sequence is restricted to start in state 1 and end in state N . Therefore

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \quad (3.39)$$

A left-to-right connected HMM topology is shown in Figure 3.4.

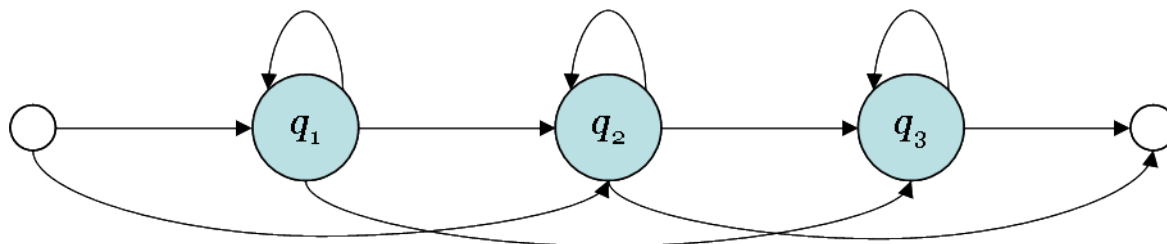


Figure 3.4: 5 State left-to-right Hidden Markov Model with non-emitting starting and terminating states

The left-to-right HMM with three emitting states are often used to model a single phoneme in speech recognition. The first emitting state represents the movement of the speech organs into position to be able to produce the phoneme. The second emitting state represents the steady-state section of the phoneme which usually has the longest duration. The third emitting state represents the movement of the speech organs away from their current position in order to produce the next phoneme. For non-vocalised plosive phonemes such as /k/ and /t/, the three emitting states represent the closing of the vocal tract as pressure is built up, the release of that pressure to produce the sound and the aspiration after producing the sound.

To provide a more flexible left-to-right structure non-emitting states are used as starting and terminating states. The non-emitting states, or null states, allow the definition of unique behaviour at the start and end of the HMM topology. For instance, the left-to-right HMM depicted in Figure 3.4 has the ability to bypass the first state and go directly from the starting state to the second state in the HMM. Similarly, the last state of the HMM can also be skipped. In fact, the left-to-right topology shown in Figure 3.4 allows the HMM to skip any state, but it can only do so one at a time. This freedom is necessary to capture the variability in different utterances of the same speech segment because humans do not produce exactly the same sequence of sounds every time a phoneme is uttered. Assimilation from neighbouring sounds and lazy speech often lead to omission of some sounds in a phoneme utterance. For example, consider the different sounds of the phoneme /t/ in the words *hat* and *tiny*. When a word ends in the sound /t/ we often omit the plosive component that is present when a word starts with it.

3.3.2 State Output Probability Distributions

The output probability distributions are representative of how the HMM models the locally stationary characteristics of the speech signal. These distributions contain the basic statistical parameters necessary to characterise data vectors that are assumed to belong to that specific state and must be chosen carefully. The objective is to choose an output probability distribution that facilitates classification, increases robustness and allows for the variability present in natural speech. The output probability distributions used in an HMM contributes greatly to the complexity and sophistication of the model, more so than the topology. Very complex statistical structures exist that can be used as output probability distributions, but we need to be careful not to allow too much freedom in the modelling process. The more complex the distributions become, the more parameters there are for the system to tweak and the more prone the system is to specialise on the training data. Overmodelling produces weak results on data the system has never encountered before and is a problem closely related to the data scarcity problem. Complex distribution structures yield bad results when training is not done on many datapoints.

Although discrete probability distributions can be used, continuous probability distributions are better suited for speech recognition applications [12]. The Gaussian distribution in its various forms, is the most popular type of distribution to use for natural language modelling because of its ability to model natural data well.

Single Gaussian Distribution

A straightforward approach would be to use a single Gaussian distribution as the output probability distribution of each emitting state. Evaluation of N -dimensional feature vectors requires a multivariate Gaussian function capable of modelling the N -dimensional space. The multivariate distribution function with N random variables is defined as

$$b_j(\mathbf{x}_t) = \frac{1}{\sqrt{(2\pi)^N |\Sigma_j|}} \exp\left\{-\frac{1}{2}(\mathbf{x}_t - \mu_j)' \Sigma_j^{-1} (\mathbf{x}_t - \mu_j)\right\} \quad (3.40)$$

where N is the dimension of the feature vectors, μ_j is the N dimensional mean vector and Σ_j is the $N \times N$ covariance matrix. The mean vector contains the mean value in all dimensions, determining the position of the distribution in the N -dimensional space. The mean is also referred to as the expected value as it has the highest likelihood of occurring. The covariance matrix contains the variance of the Gaussian distribution in each dimension on the diagonal and the correlation between the feature components of the dimensions off the diagonal. The variance of a Gaussian distribution is defined as σ^2 where σ is the standard deviation of the distribution, which is a measure of the spread of the distribution, indicating the shape of the bell curve as well as the height of the peak. In order to visualise the parameters of a multivariate Gaussian distribution function, a two-dimensional representation of feature data and the Gaussian distribution used to model it is shown in Figure 3.5.

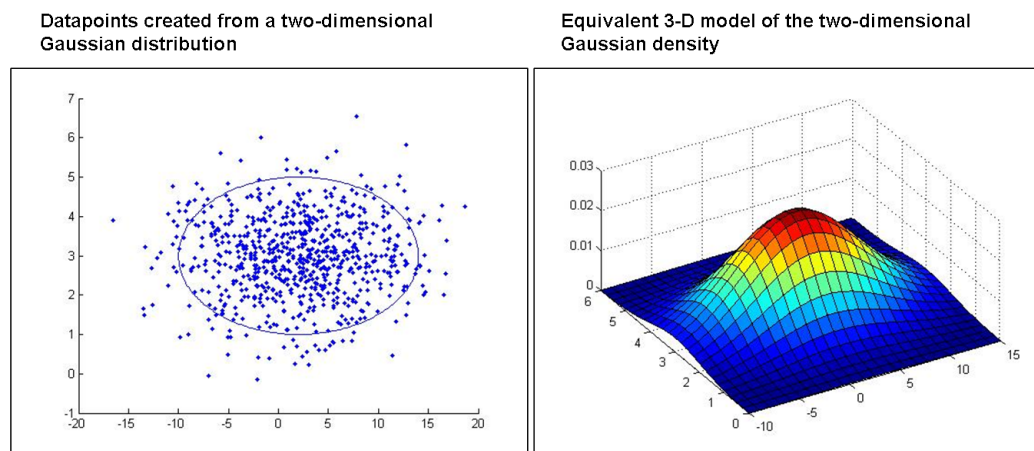


Figure 3.5: *A two-dimensional Gaussian distribution*

Regardless of the dimensionality, only second order statistical information is needed to define the mean vector and covariance matrix, thereby completely specifying the distribution function. The Gaussian distribution is very popular due to its simplicity and

efficient use of resources while still being a powerful modelling tool. The Gaussian distribution is also the probability distribution type with the lowest entropy – a numerical measure of the uncertainty of the outcome associated with a random variable. Entropy as applied to random variable distribution functions, is rooted in information theory and indicates the level of information contained in a message, or equivalently, the average loss of information if the outcome of a random variable is not known [80].

A *Full-Covariance* Gaussian distribution contains values both on and off the diagonal of the covariance matrix, representing a N-dimensional cloud of data that can take on any shape or orientation. This is the most general and powerful Gaussian distribution containing a large number of available parameters that can be trained. It also needs a large amount of training data to properly estimate these parameters, especially if the dimension of the data is high. When choosing a type of Gaussian distribution to use for a given application we always have to find a good compromise between modelling power (the more parameters to estimate, the more powerful the model) and the computational cost of working with that model (the larger the model, the more expensive it is). The full-covariance distribution is at one end of the scale and in situations where computation time and memory storage do not constrain the solution and there are massive amounts of training data available, this is the best distribution function to use.

A *Diagonal-Covariance* Gaussian distribution only contains values on the diagonal of the covariance matrix, with values off the diagonal equal to zero. The correlations between the feature components are ignored, using the assumption that the features are statistically independent and therefore uncorrelated. This greatly reduces the number of available parameters to be trained making the diagonal Gaussian distribution a good compromise between modelling power and efficient use of resources. The diagonal-covariance is robust in the face of data scarcity, especially when it is used in a larger modelling structure such as the Gaussian mixture model described below. It has been shown that the use of diagonal-covariance Gaussian distributions in speech recognition greatly reduces memory and computational requirements with a minimal loss of modelling power and recognition accuracy [12], thereby making it a very good choice for speech recognition applications.

Mixture Gaussian Distributions

Mixture Gaussian distributions are a linear set of M weighted Gaussian distributions. It is used to get a better approximation of the data by dividing the data into clusters and

modelling each independently. The M -Mixture Gaussian distribution is defined as

$$b_j(\mathbf{x}_t) = \sum_{m=1}^M c_{jm} b_{jm}(\mathbf{x}_t) \quad (3.41)$$

where c_{jm} is the weight of the m -th Gaussian distribution in the mixture Gaussian on state j . The weights of the mixture Gaussian distribution must sum to 1, making the resulting distribution a valid Gaussian distribution. Mixture Gaussians require several iterations of the expectation maximisation algorithm to obtain proper estimates of its parameters. The number of Gaussian distributions used in a mixture depends on the type of Gaussian distribution used. Full-covariance Gaussian distributions contain more parameters, needing smaller mixtures to accurately represent the data. Diagonal-covariance Gaussian distributions require slightly larger mixtures to accurately represent the data, but even with larger distributions they are still more efficient than full-covariance distributions.

Tree-based Mixture Gaussian Distributions

The drawback of having large mixtures to precisely model the input data is the considerable computation time needed to work with these models. Recently, researchers have started to develop techniques that improve the speed of the mixture Gaussian distributions by arranging them in a tree structure where each node contains an approximation of its children nodes [62]. The original set of Gaussian distributions then form the leaf nodes of the tree. When evaluating a feature vector by testing its similarity to the model, only branches that significantly contribute to the score are evaluated further, while branches with very low similarity are ignored. These structures are called Tree-based Gaussian mixture models [33]. An adaptation of this technique was developed for the pattern recognition software suite used at the DSP (Digital Signal Processing) Laboratory at the University of Stellenbosch, called PatrecII. The software was designed and implemented by Prof. Johan du Preez and Francois Cilliers and is described in detail in [18]. Tree-based Adaptive Gaussian Mixture models, or T-BAGMM, improves on the normal Tree-based Mixture models by not discarding branches of the tree with low similarity scores. Instead it keeps these approximations of the densities below it without further evaluation, while digging down deeper in branches with a higher similarity score. The T-BAGMM model therefore dynamically changes structure, modelling different regions of the feature space in different degrees of detail depending on the location of the feature vector being evaluated. The value of the T-BAGMM model lies in its speed without compromising on accuracy. The main disadvantage of using tree-based mixture Gaussian distributions is the increased memory requirement. If the tree is a balanced binary tree, which isn't always the case, double the number of densities are used than that of a normal mixture.

The acoustic models used in this research use Tree-based Adaptive Gaussian Mixture models as state output probability distributions with a maximum target mixture size of 256. The exact parameters used and the estimation procedure are discussed in Chapter 6.

3.4 Implementation of Acoustic Modelling for Phoneme Recognition

The mathematical base supplied in Sections 3.1 and 3.2, combined with the considerations of its application to speech recognition as described in Section 3.3, define the practical use of Hidden Markov Model theory for phoneme recognition. There are four phases in phoneme HMM processing that are of importance – creating the HMM structure, training the models, using the set of models to decode unknown speech and evaluating the performance of the model.

If it is known that an evaluation segment of speech is a single phoneme which simply has to be classified, it is called isolated phoneme recognition. The recognition system consists of a set of HMM models – one for each phoneme. Each HMM represents a class and during the classification process an unknown phoneme is classified as belonging to the most probable class. The unknown speech segment is converted to a set of feature vectors \mathbf{X} through the signal processing techniques discussed in Section 2.3 and evaluated for each phoneme HMM. The likelihood that the evaluation data was generated by the given phoneme model is calculated using the forward algorithm described in Section 3.2, with the phoneme model yielding the highest probability chosen as the correct phoneme class. The recognition accuracy for isolated phoneme recognition is simply calculated by noting the number of correct classifications as a fraction of the total number of phonemes tested. Unfortunately isolated phoneme recognition is not useful in continuous-speech recognition where evaluation segments of speech are sequences of phonemes of arbitrary length that must be classified without having phoneme-level boundaries available.

3.4.1 Creating and Training the Model

After deciding on the elementary subword unit to be used, an HMM model is created and trained for each of the possible classes. To train the HMM models a training data set is needed. The training set consists of feature files that contain the feature vectors obtained from signal processing, and label files that contain transcriptions and subword alignments for all speech data files in the set. To obtain transcriptions that are manually aligned at phoneme level is a very slow and expensive process, but accurate phoneme boundaries are

essential for good statistical models. Therefore bootstrapping techniques are necessary to fully utilise the training data. Words with manually aligned boundaries are equally divided into their constituent subword units with the use of a lexical decoder, which is in turn subdivided equally into the number of emitting states belonging to the HMM. These estimated class labels are far from accurate and need to be re-estimated, either with the use of generic models trained on another data set or by iteratively training models from the data, re-estimating the labels with these models and then retraining the models until a certain measure of convergence occurs. If we assume that the boundaries of the subword classes are accurate and that each feature vector in the feature file is associated with the correct class label, then the subword HMMs can be estimated using the Baum-Welch algorithm described in Section 3.2.

In creating the HMM structure we infer knowledge about articulatory and acoustic phonetics to first decide on the most appropriate HMM topology for the specific subword unit and then choose the type of state output probability distributions to be used.

The topology chosen for phonemes is usually a left-to-right structure of between 3 and 5 emitting states. The number of states impose a minimal time duration for the phoneme where each state represents a different physical vocal tract configuration. Phoneme realisations can typically be divided into three sections - the beginning contains influence from the preceding sound, the center contains the unique sound of the phoneme and the end contains influence from the following sound. Plosive consonants are the least sonorant of all phonemes. They also have the shortest durations, and usually deviate from the normal phoneme model template. However, because they also have three distinct sections – pressure build-up, plosive release of air and aspiration – they are also well modelled using a three-state left-to-right HMM.

Contextual influences are due to the nature of continuous speech where phonemes are not uttered in isolation, but assimilation and co-articulation creates a continuous flow of sounds. A typical co-articulation effect is the shortening of sounds in certain contexts, for example the phoneme /t/ in the word *later* is short in rapid speech acting as a short boundary between the two vowels. To reduce the duration restrictions and model co-articulation effects skip-links can be used to skip emitting states in the HMM.

The ability to identify the presence of silence and non-speech elements in continuous-speech recordings also require HMM modelling. Silence is characterised by little or no spectral change and can therefore be modelled by an HMM with a single emitting state. The emitting state can loop back onto itself for arbitrary lengths of silence between words and at the beginning and end of files. Non-speech elements such as coughing, throat clearing and laughter are best modelled using a fully-connected HMM topology because

of the lack of temporal structure associated with them.

The state output probability distributions are a collection of all the probability distributions associated with the emitting states of all the subword HMMs. They are considered to be subclasses of the phonemes and are usually characterised by a combination of the phoneme name and the emitting state number within that phoneme HMM. The output probability distributions can be single distributions, mixtures of distributions or even embedded hidden Markov models, creating higher-order HMMs. When training the phoneme HMMs, these output probability distributions are trained first before assembling them into the HMM structures. For each subclass the feature vectors with the appropriate labels are collected from the aligned training data and used to extract the necessary parameters to define an output probability distribution – the mean vector and covariance matrix for Gaussian distributions. Complex distribution structures require more than one training iteration to create an accurate statistical model from the data. These output probability distributions are then inserted into the HMM structures for the phoneme classes, thereby prompting additional training via Baum-Welch re-estimation.

3.4.2 Alignment of the Labelled Training Set

With no prior description of the mapping between acoustic characteristics and model parameters, the model estimation procedure is highly dependent on an accurately labelled training set. The feature vectors assigned to a specific subword unit class are used to estimate the model parameters of that specific class. Furthermore, the available manual transcriptions of hundreds of hours of speech data are usually accurately labelled on word-level, but obtaining accurate phoneme-level alignment is extremely expensive and time-consuming. The alternative is to do automatic alignment of the training data, using the lexicon to subdivide the aligned words equally into their constituent phonemes and then using pre-trained acoustic models to automatically find the most likely begin and end times of each phoneme.

Each training file is transcribed into a known sequence of phonemes. The pre-trained acoustic models representing these phonemes are chained together with an optional silence model between words and at the beginning and end of the utterance, to obtain a larger acoustic model representing the speech utterance in the training file. The Viterbi algorithm is used to obtain the optimal state sequence, given the observation feature vectors extracted from the speech utterance and the concatenated hidden Markov model for the utterance. Finding the optimal state sequence equates to labelling each feature vector with its corresponding subword class.

Of course, the effectiveness of the alignment process is directly related to the accuracy

of the pre-trained acoustic models. If no appropriate acoustic models are available for a specific data set, the alignment process, and therefore the training process, has to be bootstrapped with the use of generic phoneme models trained from large external data sets. Alternatively, the process starts by estimating phoneme alignment, equally dividing the time between accurate word boundaries. Acoustic models are trained on these rough estimates and then used to obtain better estimates on the same training data through the described alignment process. By “daisy-chaining” the process – using newly trained models as input to re-align the data and obtain better models – until the system converges, relatively accurate phoneme boundaries can be obtained. The next estimation of the model parameters is strongly affected by the segmentation obtained by the current model parameters.

3.4.3 Decoding

Decoding unknown speech data is the main purpose of a speech recognition system, utilising the statistical models obtained from training on relevant input data. For continuous phoneme recognition the phoneme models are combined into a decoder, also called a spotter, enabling the recognition of a sequence of phonemes of arbitrary length. The spotter structure is also an HMM with the phoneme models embedded in the emitting states. An elementary phoneme spotter consists of all the phoneme models in parallel with a loop-back probability to add another phoneme to the recognised sequence, thereby creating phoneme sequences of arbitrary length. This configuration is depicted in Figure 3.6.

If the transition probabilities on the initial links are proportional to the number of occurrences of each phoneme in the training set, the implication is that phonemes that rarely occur in the training set have a lower probability of occurring in the unknown utterance to be decoded. This configuration utilises a unigram grammar where the probability of a phoneme depends on its general frequency of occurrence. Alternatively, the transition probabilities can be set equal to allow the occurrence of any phoneme with equal probability, creating a model with 0-gram grammar. When using acoustic models other than phonemes, the spotter configuration must be altered to better suit that specific set of subword models.

The decoding software uses a search algorithm such as the Viterbi algorithm to find the most probable path through the spotter. States that represent the end-states of the phoneme models contain the phoneme name and are used to decode the most probable state sequence into a sequence of phonemes. In order to do word recognition, HMM models are created for each valid word in the vocabulary by concatenating the phoneme HMM models and inserting the word identifier in the end state of the word HMM. The

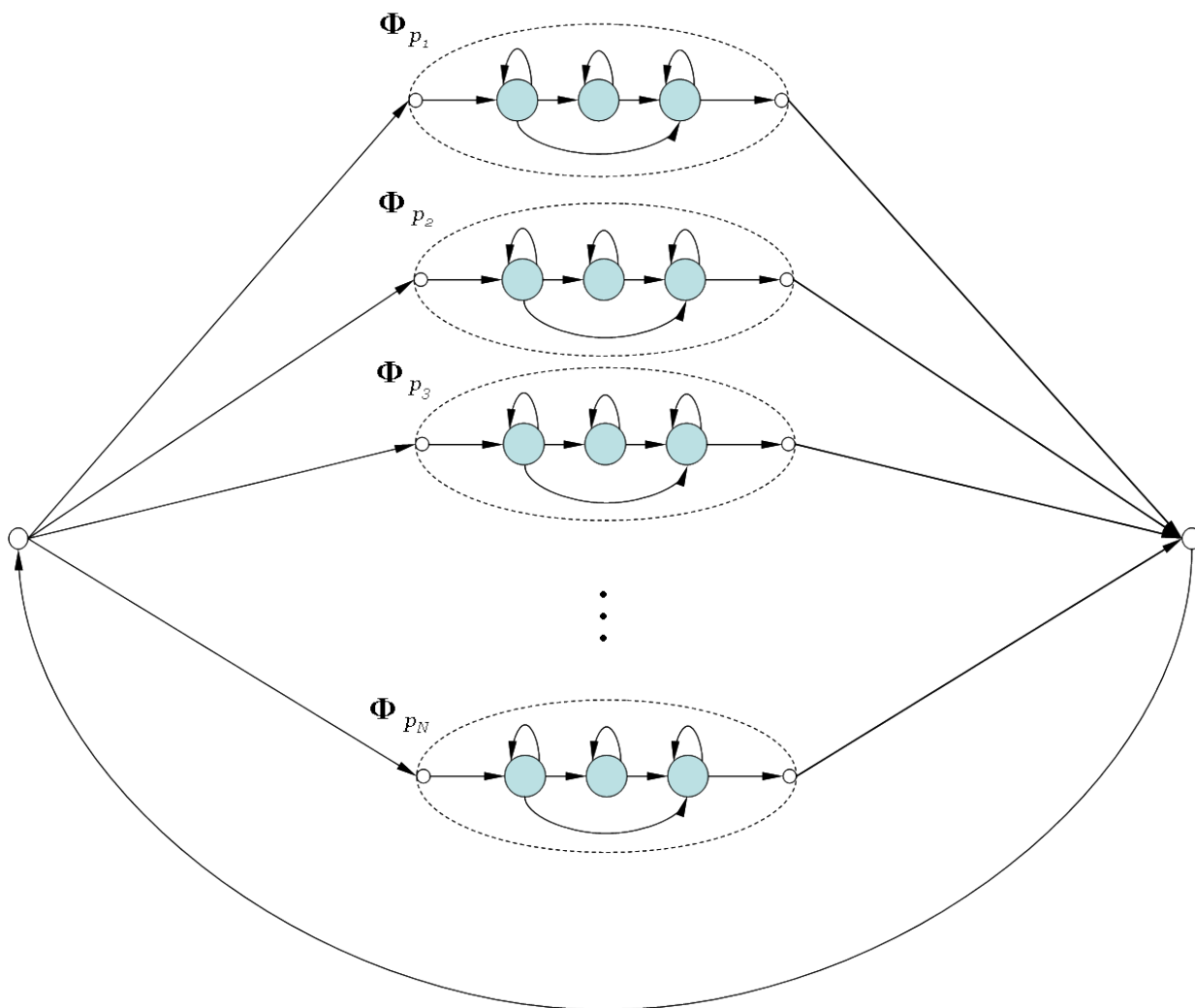


Figure 3.6: *Parallel-HMM model used for the decoding of phoneme sequences in continuous phoneme recognition*

language model is then used to connect the word models by embedding them in a larger HMM according to a certain grammar and decoding is done for the unknown speech data on the entire spotter. The HMM with estimated transition probabilities between words implement a bigram grammar where the probability of a set of two words is calculated by counting its relative frequency in a grammatical training set and using this value as the transition probability between these two words.

3.4.4 Evaluation

Phoneme recognition performance is evaluated using a testing data set with known phoneme sequences to compare to the output phoneme sequence obtained during decoding. Manual transcriptions of test files are easily obtained because subword alignment is not important, only the phoneme sequence in the test utterance is.

To determine the phoneme recognition accuracy of a continuous-speech recognition system is much more complex than for isolated phoneme recognition. In order to evaluate a decoded phoneme sequence of arbitrary length (the test sequence) it must be compared to the correct sequence obtained through manual transcription (the reference sequence). Dynamic programming algorithms are needed to align the two sequences before they can be compared. *Dynamic Time Warping* (DTW) techniques are used to warp (stretch or compress in time) the test sequence to find the optimal temporal match and the shortest distance between the two sequences. The DTW process identifies three types of errors: substitution, insertion and deletion. Substitutions occur when the time-warped sequences are compared and a correct phoneme in the reference sequence is replaced by an incorrect phoneme in the test sequence. Insertions occur when additional phonemes that do not occur in the reference sequence are added in the test sequence and deletions occur when correct phonemes in the reference sequence are omitted from the test sequence. Let NUM be the number of phonemes in the reference sequence, SUB the number of substitutions, INS the number of insertions, DEL the number of deletions and COR the number of correct phonemes. The phoneme error rate ($PhER$) can be calculated as

$$PhER = \frac{SUB + INS + DEL}{NUM} \quad (3.42)$$

with the phoneme recognition accuracy calculated as

$$ACC_{Ph} = 1 - PhER = \frac{COR - INS}{NUM} \quad (3.43)$$

The number of insertions must be subtracted from the number of correct phonemes when calculating the accuracy because the total number of phonemes used for these calculations is the number of phonemes in the reference set where the insertions are not present. The

number of insertions play a critical role in evaluating the recognition system. A system can seemingly perform well when very long sequences of phonemes are recognised because the correct phonemes are likely to be in there somewhere. It is common practise to use insertion penalties in grammars, whether on phoneme-level or word-level, to limit the number of insertions.

3.5 Summary

In this chapter the definition and theory of HMMs and their application to phoneme recognition were discussed. This represents the fundamental theory on which the mathematical implementation of acoustic modelling is built.

To be able to use hidden Markov models in speech recognition three fundamental problems and their solutions are defined. The *evaluation problem* seeks to find the similarity between an existing model and a segment of speech (observation vectors) – this problem is solved with the *forward algorithm*. The *decoding problem* seeks to find the hidden parameters of the model from a known observation sequence – this problem is solved with the *Viterbi algorithm*. The *learning problem* aims to optimise the parameters of an existing model by adjusting the parameters to maximise the probability of a given observation sequence – this problem is solved with the *Baum-Welch algorithm*. These solutions are adapted to make use of multiple observation sequences found in the training and testing data sets to both train the models and evaluate data with the trained models.

The main considerations when using HMMs for phoneme recognition are the model topology and the state output probability distributions. Different types of each were discussed and evaluated. The final part of the chapter was used to discuss different aspects of acoustic modelling using hidden Markov modelling, including creating and training the models, alignment of the labelled training set, the decoding algorithm and evaluation of the recognition system. These aspects are applicable in general and form the framework for later discussions on acoustic modelling directly related to this research.

Chapter 4

Diphones as Base Units for Speech Recognition

4.1 Speech Units Used in Linguistics

When humans decode speech signals, we recognise words while simultaneously deriving meaning from the relationships between these words. Our brains are highly efficient in piecing together separate sounds, forming words and extracting meaning from them without focusing on the individual sounds within the words.

To replicate the human perception of speech we could build a word recognition system consisting of a set of statistical models, one for each word in the vocabulary, and compare a new word utterance to each word model. Word models have the advantage of inherently modelling the fluent transitions and phonetic co-articulation present within words, but lack efficient handling of inter-word co-articulation effects. It is possible to accurately assign boundaries for words in an audio file by means of manual transcription, making the training of word models relatively easy. For speech recognition applications with a small, fixed vocabulary word-models are appropriate. However, for large-vocabulary speech recognition millions of hours of word-aligned speech data is needed for accurate estimates of all possible word models and even then the system would not be robust enough to handle previously unseen words. Therefore it lacks the ability to be generalised, which is necessary to be practical. To overcome this problem, we use a finite set of subword units contained within all possible words. Statistical models for words can then be obtained by stringing together a sequence of these subword models, as defined by a set of lexical rules.

In statistical modelling the individual acoustic models (one for each subword unit) are trained from an accurately labelled set of speech utterances. No assumptions are made beforehand about the relationships between the acoustic measurements and the subword

models, leaving the responsibility to the HMM to learn these mappings through iterative training. The resulting acoustic descriptions are therefore highly dependent on sufficient representation of all linguistically active subword units in the training data. If this is not the case there is a data scarcity problem, which is a constant concern for speech technology research.

Various types of subword units are used for speech recognition, ranging widely in complexity and computational efficiency. With fewer acoustic models the recognition system has lower complexity and higher computational efficiency due to reduced resource requirements. A small set of general models has limited modelling capability, which can reduce recognition accuracy, but because of higher class representation in the training set can also lead to better estimated statistical models. In contrast, a recognition system based on word models has a very large number of models with high complexity. Large model sets lead to low computational efficiency due to increased resource requirements. Further, dividing the datapoints in the training set into a much larger number of possible classes leads to poor parameter estimation due to data scarcity. A large number of specialised acoustic models, such as those based on words, cannot be easily generalised to handle unseen words or circumstances, whereas a small set of general models, based on monophones for instance, are more adaptable. Therefore a system based on subword units is considered the only practical solution for continuous speech recognition and has many benefits. In large-vocabulary continuous-speech recognition the choice of speech unit has a significant impact on the accuracy, complexity, expandability, and ease of adaptation to speaker or environmental variations [60].

Subword units can be classified as either context independent (CI), where the subword units are considered in isolation without regarding preceding or following sounds, or context dependent (CD), where information pertaining to neighbouring sounds are incorporated in the acoustic model.

The most prominent subword models are discussed below. Figure 4.1 shows a waveform of a speech utterance with its segmentation in terms of different subword units.

4.1.1 Syllables

Syllables are multi-phone units consisting of an optional group of consonants (the “onset”) followed by a vowel (the “peak”) and another optional group of consonants (the “coda”). In some cases the peak consists of a weak vowel or a syllabic consonant instead of a vowel. For instance the word *button* is usually pronounced /bʌtn/ in continuous speech instead of /bʌtən/, leaving the second syllable with a weak vowel. The segmentation of speech into a sequence of syllables is not definitive – disagreement is common amongst linguists

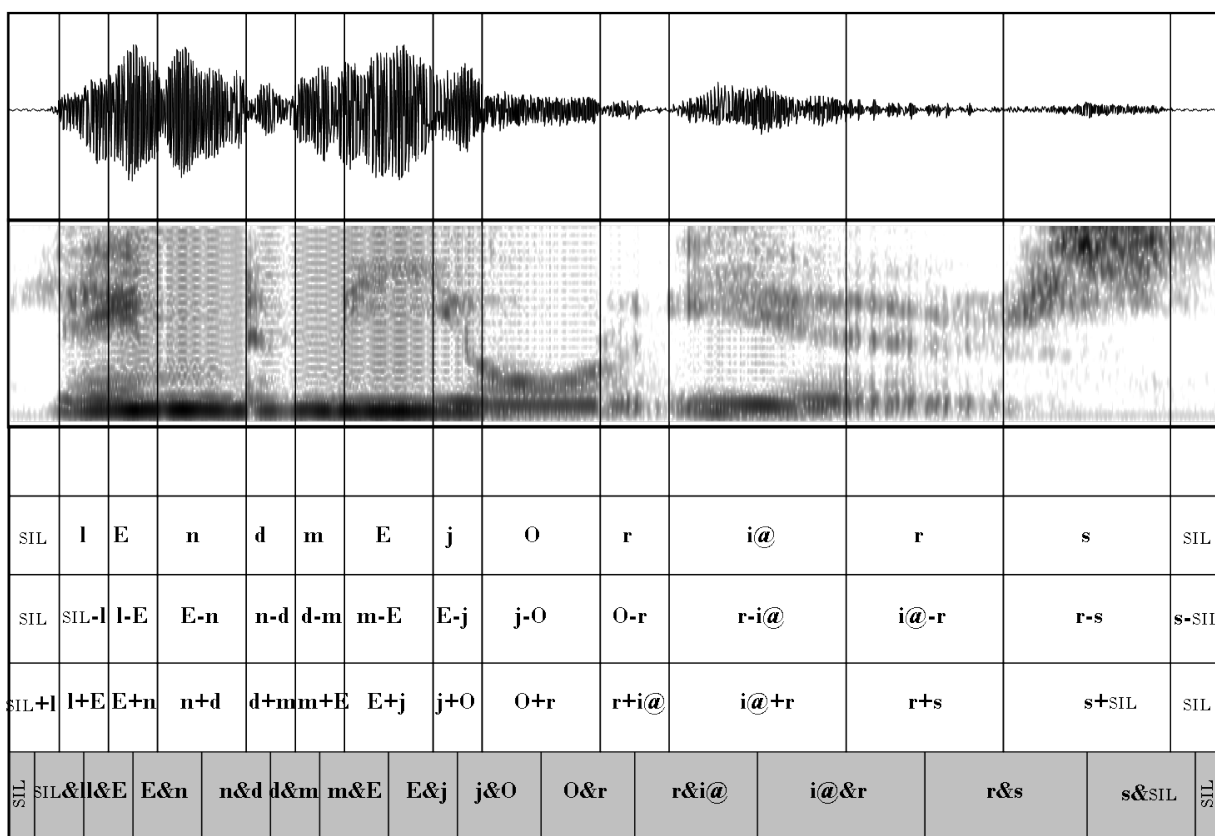


Figure 4.1: Acoustic waveform, spectrogram and corresponding subdivision into different subword units for the utterance “Lend me your ears”. The subword units from top to bottom are phonemes, left-context biphones, right-context biphones and diphones.

as to the association of consonants to one syllable or the next. They are therefore difficult to define and often differ between applications.

Syllables are considered to be the building blocks of a specific language and as such are the first sound units humans learn to produce as babies [76]. There are many restrictions as to which syllables can occur in a language, making the use of syllables for speech recognition more practical than the use of words because of the resulting finite subword model set. The inclusion of both steady-state vowels and neighbouring consonants in the syllable model means that the most important contextual information within a unit is inherently modelled. There are several hundreds of possible syllables in a language, requiring a large amount of training data for accurate statistical estimation. Therefore, although the subdivision of words into syllables results in more adaptable acoustic models than whole-word models, the large syllabic model set also leads to high complexity and low computational efficiency.

4.1.2 Monophones

The most common context independent subword units used for speech recognition are monophones – a set of linguistically defined phonemes for a specific language. Monophones are characterised by considering each phoneme individually without taking its context in a word or syllable into account and grouping all allophones of a particular phoneme together. The exact set of phonemes used to create the monophones depend on the target language and the intended use of the system. Phonemes with similar sounds are sometimes grouped together to decrease the number of acoustic monophone models used thereby increasing computational efficiency. The use of monophones as basic units for speech recognition results in N acoustic models where N is typically around 45 for the English Language. In this research we worked with the set of 44 monophones listed in Appendix A alongside the English phonemes in International Phonetic Alphabet (IPA) format.

Various linguistic phenomena such as the co-articulation found in continuous speech are ignored in favour of a simple acoustic model set. The simplicity of working with monophones is beneficial in research focused on the evaluation of speech recognition components other than acoustic modelling, such as language modelling. The advantages of using context-independent monophone models include

- easy training due to sufficient training data for the small set of different classes,
- well-developed and accessible phonetic linguistic theory,
- many available phonetic transcriptions of large-vocabulary English speech corpora,

- portability for use in new environments and contexts, such as new vocabulary sets and unseen pronunciations.

The main disadvantage of using context-independent monophone models is the relatively low recognition accuracy caused by the generic nature of the models and the inadequate modelling of the fine contextual information that is so important for accurate phoneme recognition in continuous speech.

An important consideration in acoustic modelling of monophones that can be used to counter its limited modelling capabilities, is increasing the number of densities contained in the Gaussian mixtures in each output probability density tied to a state of the HMM. Because monophones are the smallest complete subword unit set the mixture size in each state can be relatively high, compared to that of other larger subword unit sets. Data scarcity can still be a problem for some monophones that occur less frequently. To handle this situation, the number of Gaussian densities in the mixture on a specific state should be adjusted according to the availability of training data, as is the case in tree-based mixture Gaussian distributions.

4.1.3 Biphones

A biphone is a set of two adjacent phonemes where one phoneme is modelled in context of the other. Biphones can be either left-context, selecting the phonemes according to the phoneme directly preceding it, or right-context, sorting phonemes according to the phoneme it is followed by. For instance, the beginning of the phoneme /i:/ is pronounced slightly differently in the word “*we*” than in the word “*key*” because of the changes in the vocal tract needed to move from the position for the production of the consonant to the position needed for the vowel. Therefore /w-i:/ and /k-i:/ are two different left-context biphones of the same phoneme /i:/.

By differentiating between phonemes in different contexts, the acoustic models can be trained more accurately, by finely modelling subtle differences found in the different contexts. The use of basic context in acoustic modelling increases the recognition accuracy, provided that there is enough training data to accurately train each possible context dependent acoustic model. Using left- and right-context biphones also results in at most $2N^2$ acoustic models, where some contexts cannot appear in the English language. This exponential increase in the number of acoustic models is directly related to the size of the phoneme set they are based on.

4.1.4 Triphones

A triphone is a set of three adjacent phonemes where the centre phoneme is modelled in context of the phonemes directly preceding and following it. A triphone is the result of using both left-context and right-context biphones simultaneously. The increase in context modelling complexity is accompanied by an increase in recognition accuracy, provided that there is enough training data to accurately train all possible triphone contexts. The use of triphones results in at most N^3 acoustic models which requires a significant increase in resources, leading to decreased computation speed. Adaptation algorithms such as *Classification and Regression Trees* (CART) are used to reduce the total number of output probability densities through state-level tying in order to make triphone models of practical use. Many large commercial speech recognition software packages make use of triphones and pentaphones (monophones in context of two preceding and two following phonemes), thereby increasing the amount of context modelling done on the data.

4.1.5 Diphones

A diphone is a set of two adjacent phonemes where the transition between the two phonemes are modelled, usually from the middle of the first phoneme to the middle of the second phoneme. Unlike biphones or triphones, diphones incorporate contextual information by focusing on the transition between the two different sounds. These subword units are extensively used in speech synthesis (e.g. [39]) where a computer generated voice was found to sound more natural when using acoustic models that accurately model transitions, rather than concatenating quasi-stationary sounds such as phonemes. Diphone models present an attractive alternative to context-dependent modelling in solving the problem of dealing with contextual influences amongst neighbouring phonemes.

The number of diphone models are similar to that of biphones. The use of diphones results in at most N^2 acoustic models, where some transitions cannot appear in the English language. As with biphones, the exponential increase in the number of acoustic models is directly related to the size of the phoneme set the diphones are based on.

Diphones provide a good trade-off between model complexity and generality. Longer duration units such as syllables or words capture the temporal variation and influence of neighbouring sounds within their structures, but lack generality due to the large number of models needed to fully describe a given language. Monophone units are easier to work with and comprise a small set of models, but lack sophisticated modelling capacity. By shifting the unit boundaries to model phoneme transitions, diphones have sophisticated context-modelling capabilities without most of the penalties characteristic of the more

complex statistical structures.

4.2 Modelling Transitions versus Modelling Context Dependency

To maximise the potential of acoustic phonetic modelling for continuous speech it is important to consider the influence of the neighbouring phonemes of each sound, because phonemes are not produced independently. There are physical limitations to how fast the vocal articulators can move between sounds, which result in co-articulation effects. For example, in the utterance “*you too*” the consonant /t/ is placed between two rounded vowel sounds, resulting in the pronunciation of /t/ with rounded lips instead of the neutral configuration usually associated with the sound /t/ (*tea*).

These subtle changes between allophones are ignored when monophone acoustic models are used with the objective of reducing complexity and increasing computational efficiency. A common strategy to counter the loss of modelling accuracy is to incorporate co-articulation information into the monophones with the use of context-dependent models such as biphones or triphones. The aim of modelling context dependency is to capture and finely model the local acoustic variability associated with a specific sound in different contexts.

Context-dependent models have been used with great success in speech recognition systems and are widely used in current practical applications. While it is clear that the fine modelling of contextual differences in utterances of the same sound is beneficial to the accuracy of acoustic models [46], essential information could be incorporated in some other way that is less resource intensive and has lower complexity. One possible solution is the modelling of phone transitions. By shifting the focus from modelling speech sounds to modelling the transitions between the sounds, the important contextual information is captured within the models. Further, according to psycho-linguistic research [42], much of the co-articulation effects found in continuous speech can be modelled by using diphone models, although syllables are the subword units that best contain all co-articulation effects for a short segment of speech.

To investigate the potential of transition modelling, it is compared to context-dependent modelling on the basis of a few criteria: trainability, complexity, resource requirements, handling of inter-word contexts and modelling of unseen contexts.

4.2.1 Trainability

In the case of parametric modelling of hidden Markov models for transitional and context dependent models, the training challenges and considerations are similar. Given an HMM structure and labelled training data, the system automatically finds the most appropriate representation of the given subword unit, irrespective of its type. The key issues of trainability is management of the data scarcity problem and the robustness of the segment boundaries in the labelled training data.

Data Scarcity

In the face of limited training data, the higher the number of parameters to be estimated, the higher the degree of fragmentation and the bigger the problem. More complex acoustic models incorporating high levels of contextual information are often poorly estimated and statistically unreliable, leading to significantly reduced recognition accuracy. The problem of data scarcity is an ever-present one in statistical speech recognition because the collection of large amounts of accurately labelled application-specific training data is very expensive and time-consuming. Both transition models and context-dependent models commonly suffer from data scarcity due to their relatively big set sizes.

When working with a limited amount of training data the most important consideration is how to fully utilise the little data available to maximise the recognition potential of the system. It is very important to choose the optimal complexity of the acoustic models that can provide the best possible starting point to model the current training set. This choice should provide a trade-off between simple, well-estimated models (few parameters) and complex models (many parameters) that incorporate additional contextual information. The extent of the data scarcity problem determines the acoustic model complexity that can be handled without severe loss of reliability of the statistical models.

The balance can be achieved by starting with complex acoustic models and reducing the total number of model parameters through techniques such as decision tree based state clustering (CART trees) and/or adjusting the types and sizes of the output probability Gaussian mixture models.

Additionally, context-dependent models can utilise a form of regression to fall back on acoustic models with less context modelling capabilities when data representation falls below a certain threshold for a specific class. For example if the triphone $p_L - P + p_R$ has insufficient representation in the training set, the combination of left context biphone $p_L - P$ and right context biphone $P + p_R$ can be used instead. In the worst case, the triphone set is reduced back to the monophone set. Transition models cannot utilise regression techniques, but they can be built from well-trained monophone models when

the available training data is insufficient. In the worst case, if no diphone is sufficiently represented in the training set, the diphones are then effectively reduced to the monophone set when they are combined again.

Robustness of Segment Boundaries

In an aligned sequence of phonemes decoded from a speech utterance, a single phoneme segment represents a portion of the acoustic signal that is considered a sound separable from the surrounding sounds. Although the phoneme sequence is not a concatenation of steady sounds but a continuous flow from one sound to another, the central portion of each phoneme is approximately stationary, at least for a short period. The phoneme sequence can therefore be seen as a sequence of target articulatory configurations or sounds with relatively long transitional periods between these steady-state components.

When continuous speech is segmented into its constituent phonemes, the boundaries between these phonemes can be difficult to pin-point. Therefore phonemes, and context dependent subword units based on phonemes such as biphones and triphones, are very sensitive to accurate alignment of the training data. The exact boundary location where the transitional characteristics are subdivided into the two broad phoneme categories, can have a considerable impact on the estimation of the model parameters. The use of diphones as an alternative subword unit in early research was largely driven by a need for robust segmentation [79].

In contrast, boundaries between segments of transitional models are expected to be situated in the relatively steady-state (central) portion of the phoneme. The segmentation boundaries are therefore more robust and the transitional model parameters should not be affected that much by small changes in boundary locations. This hypothesis was proved to be correct by Dobrisek *et al.* [27]. Placing subword unit boundaries in a relatively stable region also improves fluidity when neighbouring diphones are concatenated.

It was also shown in [27] that if biphones are automatically aligned from word sequences and trained from scratch instead of being initialised with monophone boundaries, the biphone models approximate diphone models. The model estimation and subsequent data alignment shifts the biphone segment boundaries to more closely approximate diphone boundaries. This suggests that diphones characterise speech segments that are more distinguishable and separable from each other in the feature space when compared to biphones.

If we analyse the characteristics of each state of a three-state left-to-right hidden Markov model for a left-context biphone, triphone and diphone under the assumption that they have an equal number of parameters, certain observations can be made. These

observations are summarised in Table 4.1.

Table 4.1: *Comparison of modelling characteristics on state-level between a left-context biphone, triphone and diphone model in similar parametric conditions. All models are assumed to be three-state, left-to-right hidden Markov models.*

Left Emitting State	Central Emitting State	Right Emitting State
Left-context Biphone		
Very specific to a single fixed context	Specific; Situated in static central portion of the phoneme; Fixed left-context and varying right-context	Very general; Used to model all contexts following this biphone
Triphone		
Specific to a fixed context, but clustered with other contexts for parameter reduction	Very specific; Situated in static central portion of the phoneme; Fixed left- and right-context	Specific to a fixed context, but clustered with other contexts for parameter reduction
Diphone		
Very specific; Left-context situated in static central portion of the first phoneme; Right-context specific to a single diphone transition	Very Specific to a single diphone transition	Very specific; Right-context situated in static central portion of the second phoneme; Left-context specific to a single diphone transition

Biphones that characterise the same phoneme in different contexts, differ only on the side of the phoneme of which the context is modelled. For instance, in the case of a three-state, left-to-right left-context biphone, the left-most emitting state is unique to the current context. The central emitting state models the stationary part of the phoneme. Its start is relatively unique to the context and its end is relatively generic, as all the right-contexts are grouped together. The right-most emitting state is very generic because it has to be common to all left-contexts of the current phoneme. In contrast, for a three-state left-to-right diphone, the central emitting state is unique to the current transition. The left-most and right-most emitting states are both specific to the current transition on the side closest to the central state, while modelling the stationary part of the phonemes

on the outside. Ideally the acoustic model should aim to closely characterise the class-specific information in as few parameters as possible, thereby minimising the number of parameters that model generic characteristics. Hence diphones are better suited to accurately model co-articulation information with a limited number of parameters than biphones. Diphone models contain more parameters that are free to be trained to represent characteristics that are unique to certain transitions.

4.2.2 Complexity and Resource Requirements

In an ideal research environment with unlimited resources in terms of time, hardware and input data, high complexity acoustic models with maximum context-modelling capabilities will always outperform the simpler systems. However, practical systems are often severely limited regarding the resources available on the user platform. Market requirements can be quite severe in terms of computational complexity and memory space. It is therefore desirable to be able to maximise acoustic modelling ability and speech recognition accuracy within the framework of limited resource environments.

To find a good balance between model sophistication and resource requirements, several techniques are used to fine-tune the number of model parameters, the most common of which reduce the total size of the acoustic models through Gaussian distribution merging such as decision-tree based state clustering [85]. Another common technique is to vary the complexity of the state-level distributions. The state output probability density functions are logically clustered to produce an acoustic model set with fewer unique parameters to be estimated, but with minimal loss in accuracy. The available training data is used to estimate the optimal size for each Gaussian mixture probability density on the HMM states, based on the number of datapoints available for a specific class. These techniques include the use of simple covariance structures for the Gaussian densities (diagonal covariance versus full covariance) and the use of a small number of mixture components for each Gaussian mixture probability density. Optimal selection of these modelling criteria can be determined dynamically by optimising a Bayesian information criterion [15].

In terms of the total number of parameters to be estimated, biphones are roughly twice as large as diphones if both left- and right-context biphones are used, triphones are exponentially larger and pentaphones cause a parameter set explosion. Unlike context-dependent models, transitional models optimally position each parameter to capture the essential transitional information with the lowest possible number of parameters. The transitional models therefore have a better starting point than context-dependent models when parameter reduction techniques such as decision tree based state clustering is used to reduce the parameter set.

4.2.3 Handling Inter-word Contexts

The handling of inter-word contexts or transitions is problematic for both context modelling and transition modelling. As soon as contexts across, or transitions between words are to be modelled, the lexicon can no longer be used to derive a logical sequence of subword units from a sequence of words. This is because the lexicon is limited to descriptions of isolated words and their intra-word contexts or transitions. In many systems, inter-word characteristics are therefore either left out altogether or left to the language modelling portion of the speech decoder.

Context-dependent models can use regression techniques to utilise more generic models such as monophones or left- and right-context dependent models at the beginning and end of each word. These smaller models are also necessary at the beginning and end of each speech utterance to tie in the endpoints of the label sequence.

Transition models also need context-independent monophone models at the beginning and end of each speech utterance. These models can also be used between words during language modelling, but may subtract from the diphone influence, especially in utterances containing many short words. Another possibility is to bypass the use of a lexicon by translating a decoded sequence of transition models into the equivalent monophone models before continuing on to word recognition applications. Lexical structures designed to handle inter-word contexts when using diphones have been used with great success [32]. These structures typically define multiple paths at the junction of two words, one directly modelling the transition at the boundary of the words and another one including transitions to and from an optional silence.

To isolate the influence of the different acoustic models from the influence of language modelling, only phoneme accuracy was measured in this research. Transition sequences or context-dependent sequences were translated to their equivalent phoneme sequences after decoding. Because handling of inter-word contexts is essentially a language modelling problem, it was not explicitly addressed.

4.2.4 Modelling of Unseen Contexts

The difficulties related to data scarcity and insufficient class representation for model sets with a large number of parameters continue on to the problem of modelling unseen contexts. If certain valid context-dependent models or transitional configurations did not occur in the training set, but do occur in the evaluation set, the system will either automatically misclassify a portion of the speech signal or will not be able to handle the situation gracefully.

For both context modelling and transition modelling regression techniques can be used to initialise unseen contexts or transitions from a more generally trained statistical model such as one based on monophones [14, 10]. But not all possible acoustic contexts or transitions are valid in a specific language. Without the availability of a sufficiently representative training set, there is no way of knowing which missing contexts can still possibly occur during testing and which cannot. The additional resources needed to maintain a set of general building blocks can affect the system negatively, therefore merely constructing all missing contexts should be avoided. External linguistic knowledge can be used to list all possible contexts or transitions within a language from which items missing from the training set can be constructed from monophone building blocks. These models should then be flagged as “untrained” models during testing to ensure that they are only utilised when needed, giving preference to the models trained on the training data.

Alternatively, unseen contexts or transitions can be handled during the testing phase. When all models within the set of trained models yield a sufficiently low score, the decoder can experiment with combinations of the monophone building blocks to determine whether this segment is an unseen context. This solution can, however, significantly increase computational complexity and slow down the decoding process.

4.3 Implementation Strategies for Diphones

There are a few descriptions in published research of different implementation strategies for the use of diphones in speech recognition. The literature study provided in Section 2.2 details research done on diphone-based recognition systems over the past few decades. To objectively view HMM-based systems with diphones as subword units, implementation strategies commonly used with diphones are discussed below. This section deals primarily with technical details and will refer back to Section 2.2 where appropriate.

4.3.1 Non-parametric Methods

Non-parametric methods do not rely on a given probability distribution function to characterise the input data. The statistical description of the underlying process is driven by the data, therefore large amounts of data are needed to confidently make assumptions about the process. Non-parametric methods can be powerful statistical tools, especially in applications where it is beneficial to restrict the use of assumptions about the underlying process being modelled. Non-parametric methods also use fewer parameters than parametric modelling, sidestepping the issue of poor parameter estimation in complex statistical models.

Template Extraction

Template extraction was one of the first implementation strategies to be used with diphones. A dictionary of templates are created where each model characterises the transition inherent in a diphone as a “typical pattern”. These templates are either manually created through careful analysis of the spectral characteristics in the speech data or automatically extracted with the aid of bootstrapping algorithms. During decoding the recognition system continuously measures the similarity of a portion of the incoming speech pattern, centered around the present time interval, with the complete set of diphone templates. A similarity score based on maximum likelihood is calculated for each template and evaluated to find the most likely diphone sequence [78].

Diphones were commonly used as templates because they are the smallest set of subword units that take co-articulation effects into account [79]. Often steady-state models and transitions spanning more than two sounds are included in the template dictionary to collectively form “diphone-like” units. Diphone templates were initially used only in experiments with very small vocabularies, such as spoken sequences of digits, to simplify the recognition process, because small vocabularies result in a small set of valid transitions. The template dictionaries are therefore small enough in order for each transition to receive individual attention. This manual approach is not practical in large vocabulary speech recognition, which requires the means to extract the diphone templates automatically. Automatic diphone bootstrapping have the added ability to make the system speaker-adaptive – the system can be used to adjust generic models to create individual models for a specific speaker.

To make use of popular and successful HMM-based algorithms, the HMM/template hybrid system combines the detailed modelling capabilities of the templates with the solid mathematical framework of an HMM system for reliable model training and parameter estimation. The output probability density on each state of the HMM is replaced by a template structure representing a typical sequence of observations characterising the short segment of speech represented by the state. Dynamic time warping algorithms are used to measure the similarity between the model templates and observation vectors from the unknown signal. Using templates as non-stationary states in an HMM addresses an underlying assumption in HMM-based systems – observation vectors pertaining to the same HMM state are independent and identically distributed (IID assumption). The IID assumption, although reasonable, is not completely true as successive observation vectors extracted from speech are statistically dependent and non-stationary. Using a piecewise linear set of templates can address this issue while still maintaining the mathematical stability and simplicity of hidden Markov models, by using adapted HMM model estimation

algorithms. An example of this implementation can be found in [36].

Multi-trajectory Subspace Models

Subspace models rely on a temporal series of feature vectors to visualise and model trajectories of important speech dynamics. This method is derived from the algorithm for finding *Principal Curves*, first described in 1989 [40]. An adaptation of the well-known technique for dimensionality reduction, Principal Component Analysis (PCA) is used to optimally reduce the dimensionality of the feature vector space while still preserving the temporal ordering of the data sequence. This technique is known as time-constrained PCA (TC-PCA). A trajectory template is created for each diphone model using a maximum likelihood criterion based on the EM algorithm. To obtain an evaluation score a test trajectory is time warped to match the length of the template after which a distance measure can be calculated.

Subspace models are ideal for capturing the transitional information within diphones and to maximise discrimination between sounds that are acoustically very similar, such as /b/ and /d/ [69]. Subspace models have the ability to finely model spectral changes whereas other methods make crude approximations, for example hidden Markov models that use piecewise constant approximations. By focusing on transitions and not on steady-state portions of the speech signal, the acoustic models are more distinctive and aid classification. Studies on the use of multi-trajectory subspace models have shown improved results over similar systems based on monophone models if dimensionality is excessively reduced [70, 72]. For subspace models of two dimensions, HMM systems outperform subspace models but at a higher cost in number of parameters [71]. Similar to templates, subspace models can be used in conjunction with HMM-based systems to improve overall recognition accuracy [73].

4.3.2 Parametric Methods

In contrast with non-parametric methods, parametric statistical methods assume that the observation data is created from and can be completely described by some probability distribution function [48]. The observation data is summarised into a finite set of parameters such as the mean value and spread of the data distribution. Use of parametric methods such as hidden Markov models and neural networks is very common in speech recognition technology. The main consideration of parametric models is reliable parameter estimation, which greatly influences system performance and usefulness.

Neural Networks

An artificial neural network (ANN) is a mathematical model, first conceived in the 1940s, based on examination of neurological functioning. The idea was to create artificial intelligence that closely resemble information processing in the human brain. The ANN consists of a network of interconnected nodes (neurons) often organised in layers (multi-layer perceptrons). However, modern software implementations tend to move away from the biological roots of ANNs towards efficient implementation of non-linear, distributed, parallel and local processing and adaptation [1]. Each node in the neural network contains a mathematical function for determining – based on the node inputs – whether or not the “neuron will fire”, propagating the signal onto the node output. During training, connection weights are adjusted to yield the expected result of each set of training inputs.

Artificial neural networks are commonly used in conjunction with hidden Markov models to incorporate temporal modelling capabilities. State output probability densities in the HMM structure are replaced by neural networks designed to model piecewise linear portions of the speech signal. However, hybrid HMM/ANN systems have more parameters to train than regular HMM systems, increasing computational and storage requirements and risking poor parameter estimation if limited training data is available.

Hidden Markov Models

Most modern speech recognition systems use hidden Markov models as their implementation of choice. It therefore makes sense that a large portion of research done on diphones for speech recognition is based on HMM systems. Basic HMM theory is discussed in Chapter 3 and prominent diphone-based HMM-systems are discussed in Section 2.2. Similar to monophones, diphones are modelled as left-to-right, continuous density hidden Markov models with Gaussian mixtures as output probability density functions. Standard training and evaluation algorithms apply.

A hidden-articulator Markov model (HAMM) is a special type of HMM where each state represents a physical articulatory configuration instead of a statistical model for feature vector data. A detailed discussion of the HAMM recognition system can be found in [74]. Articulatory knowledge suggests that the speech production system consists of a fixed number of articulators, each of which can only be in one of a finite number of positions. Traditional feature vectors are therefore replaced with vectors describing the numerical position of each articulator. A hidden Markov model is trained for each diphone unit by manually noting the starting and ending configurations and filling in all possible routes to get from start to finish. Transition probabilities are trained by means of the Baum-Welch algorithm.

4.3.3 Automatic Diphone Segmentation

Research has been done on automatic segmentation of speech signals into a sequence of diphones by finding the areas where large amounts of temporal variation occur [4, 83]. Defining a parameter for spectral feature variation quantification allows the identification of areas with high rates of change, each of which can then be defined as the centre of a diphone. Automatic segmentation can lead to more natural boundaries when the feature variation parameter is analyzed correctly, but finding the optimal diphone boundaries may not be as easy as noting the local maxima of the parameter. Without the framework of an existing phoneme segmentation, the automatic diphone segmentation can yield classes that are not standard diphones in the target language, especially when analyzing continuous speech. Transitions that appear fluent across more than two phonemes may also be grouped together. The inclusion of these classes can therefore lead to a more general class set of sound transitions that may be much larger than the set of valid diphones in the language.

Another novel approach to automatic segmentation is to align transcriptions to their corresponding speech signals using a speech synthesiser [52]. The segmentation process is done by creating a high-quality synthetic reference speech pattern and comparing it to the actual speech pattern. This approach has the advantage that no pre-training stages are needed to create accurate segment boundaries. The high-quality speech synthesizer is based on diphone units and therefore perfect for diphone-based segmentation.

4.4 Acoustic Modelling with Diphones as Base Unit

Training hidden Markov models for diphones presents a series of challenges that need careful consideration. The paradigm shift from modelling quasi-stationary sounds to modelling the transitions between these sounds leads to changes in all aspects of the recognition system:

- segmentation of speech signals into subword units,
- the HMM topology and nature of the output probability distributions used for the models,
- the interaction between these models during the decoding phase and
- the handling of lexical and grammatical structures used in the recognition of word sequences.

4.4.1 Segmentation

Acoustic models trained on a labelled database is extremely sensitive to the accuracy of the segmentation boundaries, as incorrectly labelled frames of speech can cause severe deterioration in system accuracy. Accurate subword segmentations can be obtained through manual transcription, but this process is time-consuming and very expensive, especially for the large volumes of data used to train models for large-vocabulary continuous-speech systems. Various complex automatic segmentation algorithms have been developed to provide quick estimates for segmentation boundaries.

A simple alternative is to iteratively improve segmentation boundaries by alternately training acoustic models from the segmented data set and re-aligning the data set with the new model set. The initial boundaries can be rough estimates based on word-level boundaries and refined through a daisy-chaining process. The AST data set used in this research was orthographically transcribed and accurately aligned on phoneme level, providing a very good starting point for further segmentation to sub-phoneme level. The sub-phoneme segments represent the individual HMM states with first approximation boundaries equally dividing the phoneme segment into the number of emitting states. In this research one iteration of phoneme model training and training set realignment was typically done to obtain segmentation boundaries on sub-phoneme level. For diphones, the phoneme-aligned data set was used to create diphone boundaries – from the middle of one phoneme to the middle of the next phoneme. The sub-diphone boundaries are then obtained in a similar manner as the method used for sub-phonemes, depending on the HMM topology chosen for each diphone model.

4.4.2 Model Structure

The HMM topology used for diphone models in this research is similar to the topology used for monophone models. This is due to the fact that no change in the basic left-to-right topology would improve the modelling capacity for diphone characteristics. The transitional models are still considered a temporal sequence of quasi-stationary events with probabilistic transitions between them.

Initial diphone experiments were aimed at providing a first approximation for diphone models by using the same methodologies as used with monophone models, but with the new segmentation boundaries. These experiments used 3-state left-to right models with one skiplink on each state for each diphone model. This configuration results in the first emitting state modelling the last half of the previous phoneme, the last emitting state modelling the first half of the next phoneme and the central emitting state modelling the

critical transition between the two. Modelling the transition with a single state, which assumes stationarity, is not the best implementation of transition modelling. A better idea would be to use more than one central emitting state, but that leads to larger statistical models with more parameters to be estimated.

Subsequent diphone experiments used monophone models to construct the diphone models from and adopted the monophone topology. Diphones that transition to or from silence, use a single state to represent that portion of the model. Similarly, diphones that transition to or from non-speech segments, use a 3-state fully connected topology to represent that portion of the model. Transitions to or from regular phonemes use left-to-right topologies according to the number of emitting states usually associated with each phoneme. The result is diphone models with different numbers of emitting states and hybrid HMM topologies.

There are many valid diphone models that can occur in any given language, making individual attention to HMM topology difficult, but a diphone recognition system can certainly benefit from making careful decisions for each possible transition. Transitions involving plosive consonants will contain clearly defined sections (pressure buildup, burst and aspiration), whereas transitions involving diphthong vowels will require longer sequences of emitting states to accurately reflect the more gradual gliding changes. These considerations were not taken into account for this research, as the simpler solutions described above was considered to be more appropriate in conditions where resources are limited.

4.4.3 Decoding

The transitional characteristics inherent in diphone models, lead to natural restrictions as to which diphone models can follow a given diphone. These restrictions lead to a slightly varied format of the spotter used to decode a speech signal into a sequence of class labels. This spotter format is explained in Section 5.2. All other decoding algorithms are exactly as described in Section 3.4. The result is a sequence of diphone models that logically fit together to recombine “phoneme halves”.

4.4.4 Evaluation

Phoneme Recognition Accuracy

The evaluation procedure described in Section 3.4 is used for continuous phoneme recognition and consists of comparing the phoneme sequence obtained from the decoder with a correctly labelled reference sequence. When using diphone models for phoneme recog-

dition, the diphone sequence obtained from the decoder is first transformed into the equivalent phoneme sequence. This constitutes the reverse of the process used during segmentation to obtain diphone labels from monophones. The diphone sequence is assumed to start and end with silence, modelled as “half a diphone” at both ends. The conversion from diphone labels to phoneme labels is straightforward if the correct decoder was used to avoid consecutive mismatched diphones.

Once the equivalent phoneme sequence is obtained, it can be evaluated against the reference sequence with the same algorithms normally used.

Phoneme recognition accuracy can be used as an accurate measure of the acoustic modelling capability of the current speech recognition system. Practical systems, however, usually continue on to language modelling to obtain word sequences from the speech signals.

Word Recognition Accuracy

To decode word sequences, subword models are usually combined according to a set of lexical rules to create word models, which can then be interconnected using language model constraints. To mimic this procedure with diphone models, requires the handling of inter-word contexts where two words are connected, as discussed in Section 4.2.3. Language modelling and subsequent word recognition were not considered in the work done for this thesis.

4.5 Summary

In this chapter an in-depth analysis of diphones as subword units for speech recognition is done. To be able to place diphones in context with other popular subword units, the characteristics, advantages and disadvantages of using syllables, monophones, biphones, triphones and diphones are discussed. These subword units can be broadly categorised as being either context-independent (only considering isolated phonemes) or context-dependent (incorporating neighbouring phonemes and contextual information into the current model). Diphones are a special type of context-dependent model – a transitional model.

The next part of the chapter investigates the differences between transition modelling and context-dependent modelling in terms of criteria such as trainability, complexity, resource requirements, handling of inter-word contexts and modelling of unseen contexts. Both transition models and traditional context-dependent models aim to model the important co-articulation information present in continuous speech and both types suffer

from the problem of data scarcity due to the exponential increase in the total number of models, which leads to low class representation in the training data and subsequently to poorly estimated model parameters. Parameter smoothing and tying techniques are commonly used to reduce the parameter set size and increase recognition accuracy. Lower-level context-dependent models with higher class representation can be used through regression techniques to handle unseen contexts and inter-word contexts.

One area where transition models have an advantage over traditional context-dependent models is the robustness of segment boundaries, due to the fact that the model boundaries are situated in the relatively stationary portion in the centre of a phoneme. Therefore, small changes in boundary location does not have the effect it would have on the corresponding context-dependent boundaries situated in the fast changing transition zone between phonemes. Another interesting observation is that biphone boundaries, when automatically segmented and trained to find the correct positioning tend to move towards diphone boundaries. Apart from different naming conventions, diphones and biphones will essentially result in the same segmentation of a labelled training set if no prior information, such as the location of monophone boundaries, is known. This suggests that diphone segmentation will lead to classes that are better distinguished than biphone segmentation based on monophone boundaries.

In Section 4.3 different implementation strategies for using diphones or other transitional models in speech recognition are discussed. The different techniques employed by researchers reveal much about how each of them find a good way to utilise diphone characteristics for speech recognition.

In order to use diphones as subword units in acoustic modelling, many of the considerations discussed in Chapter 3 are adapted. These changes and considerations, including segmentation, model structure, decoding and evaluation are discussed in the last part of the chapter.

Chapter 5

Adaptation Techniques for Diphone Models

Model adaptation techniques are commonly used to enhance or adapt existing models for use in task-specific situations. Some adaptation techniques aim at better utilisation of the input signal, such as noise compensation or channel adaptation, while other techniques are used to adapt generic models trained on external data to the current application conditions, such as speaker adaptation, when little training data is available. The model adaptation techniques discussed in this chapter cover a broad range of methodologies used to create better diphone models and are essential to the success of diphones in speech recognition. This is because dealing with diphone models in the same way as monophone models introduces a range of problems related to high system requirements and data scarcity. The number of models are increased exponentially (1600 diphones from a set of 40 monophones), leading to higher memory requirements, a slower recognition system and a more complex decoder. Also, when training more models, the representation of each category in the training set is reduced, often below the level needed for accurate parameter estimation. This problem needs to be addressed without simply acquiring more training data. In this research these challenges are addressed in part by utilising a range of adaptation techniques specifically aimed at creating a fast, accurate diphone-based phoneme recognition system. The experiments described in Chapter 6 demonstrate the contribution of each adaptation technique described in this chapter to the final system.

5.1 Diphthong Splitting

The maximum number of possible combinations of ordered phoneme pairs is defined as the number of monophones squared. Although not all diphones are used in a given

language, the exponential increase in the number of parameters to be estimated render the basic diphone-based phoneme recognition system impractical for use in limited resource environments, especially when the available training data is limited as well. The system needs a significant reduction in the number of parameters that are to be estimated and the fastest, simplest way to do this is to use a minimised set of monophones as a starting point.

Most languages contain diphthongs, a sequence of phones consisting of one vowel gliding to another that is usually interpreted as a single vowel. The English word *loud* contains the diphthong /*au*/ that is equivalent to the transition between the two vowels /*a*/ and /*u*/. Affricates are often grouped with diphthongs due to the fact that they are also transitions between sounds that are considered phonemes in their own right. The two affricates in English are /*tʃ*/ as in *chips* (combination of phonemes /*t*/ and /*ʃ*/) and /*dʒ*/ as in *job* (combination of phonemes /*d*/ and /*ʒ*/). The English language contains between 8 and 10 diphthongs, depending on the monophone set used. By splitting these diphthongs into their constituent phonemes, which are already present in the monophone set, instead of considering them as additional phonemes, the monophone set is reduced by roughly a quarter.

Diphthongs are usually included in the original monophone set as additional vowel sounds because the important information pertaining to the diphthong sound, the sliding transition between the vowels, cannot be characterised by two separate monophone models. However, diphones are designed to model the transition between phones. The transitions between the vowel sounds can be explicitly modelled, given that the constituent vowels are already in the set of monophones. Therefore, splitting the diphthongs into their constituent vowel sounds will result in no loss of information or modelling capability.

The experiments done in this project were based on a set of 44 monophones for South African English. The complete monophone set in International Phonetic Alphabet (IPA) notation and the equivalent symbols used in the experiments are listed in Table A.1. From 44 monophones a diphone set with a maximum size of 1936 can be created. By using diphthong splitting, the monophone set is reduced to 34, producing a diphone set with a maximum size of 1156 models – 60% of the original set.

5.2 Basic Diphone Grammar for Phoneme Spotting

Language models and grammars are used to decode word sequences from speech signals as part of the speech recognition process. The monophone models are strung together with the aid of a lexicon to create word models and these word models are combined

according to linguistic rules to define valid word sequences. In the absence of language models, phoneme recognition is used to decode phoneme sequences from the speech signal. A simple grammatical structure is used as a decoder to signify the valid transitions from one phoneme to another, the simplest of these being a *0-gram* spotter that has the same topology as shown in Figure 3.6. The 0-gram spotter is used to decode phoneme sequences with arbitrary length where any phoneme can be followed by any other phoneme with equal probability. Phoneme recognition is often used to evaluate acoustic models, isolated from the contributions of language models.

If a 0-gram spotter is used, an increased number of subword models in the spotter (all equally likely to occur) naturally causes a decrease in recognition accuracy. In the case of monophones the spotter has to choose the best candidate from a set of 44 models, but in the case of the diphones the spotter has to make its choice from a set of 1936 models. Fortunately the diphone structure lends itself to the creation of a better spotter structure for phoneme spotting – a basic diphone grammar.

Diphones are essentially the last half of one phoneme together with the first half of the next phoneme and as such can only be followed by diphones starting with the last half of the second phoneme. They fit together like dominoes. For instance, the diphone “p&E” cannot be followed by the diphone “ae&t”, but “E&t” and “E&k” are valid next diphones. These transitional constraints are accompanied by the incorporation of monophone models to handle “partial diphones” at the beginning and end of the diphone sequence. To preserve generality, the sequence must be able to start and end with any phoneme, which requires the inclusion of all monophone models into the set of diphone models to be used as partial diphones. However, in the case of continuous speech utterances it is not unreasonable to assume the presence of silence at the beginning and end of each recording, even if it is only a few frames that would be imperceptible to the human ear. By restricting the diphone sequence to start and end with silence, only one monophone model is needed. The silence model used in the diphone spotter for the beginning and end of each utterance is a well-trained monophone model, only used during the decoding phase. Figure 5.1 shows a diphone spotter utilising transitional and end-point constraints to create a basic diphone grammar for a reduced hypothetical monophone set.

Including constraints unique to the characteristics of the diphone models into the spotter greatly improves recognition accuracy, but the underlying grammar is still a 0-gram grammar. The transitional probabilities between diphones in the spotter model are all equal with no additional information used as to the frequency of occurrence of each diphone.

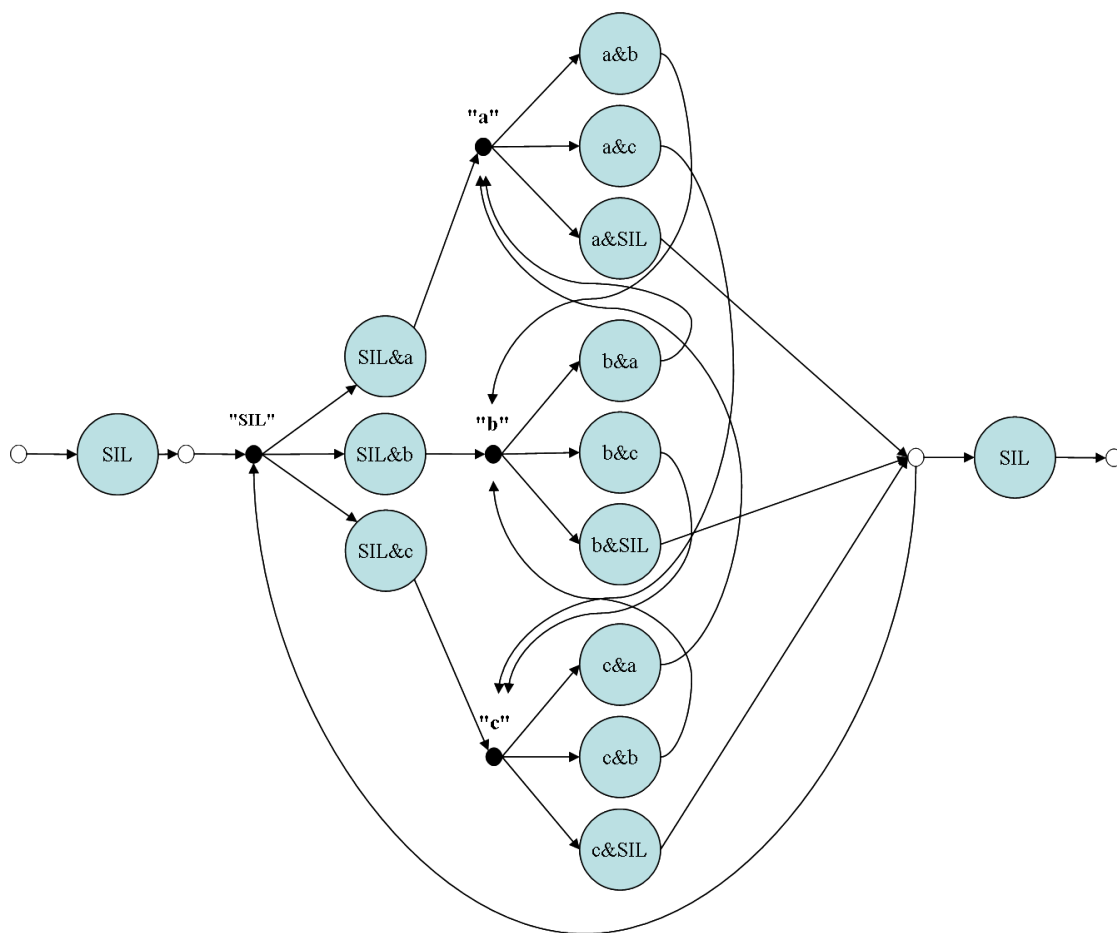


Figure 5.1: Basic diphone grammar for the language defined by monophones “SIL”, “a”, “b” and “c”. The spotter assumes the existence of silence at the beginning and end of each utterance. The black null states are called cluster states and represent the set of diphones with a common first monophone.

5.3 Diphone Set Completion

Depending on the size of the training data set, the set of valid diphones trained from it may be incomplete due to the omission of some rare diphones. From Table B.1 it can be seen that there are some phonemes that occur very rarely in the training data. Diphones modelling transitions to or from these phonemes will be even more rare and can lead to some diphones being absent from the training data set altogether, but encountered in the testing data set. The lack of sufficient data to train these diphone models presents us with an interesting dilemma – omit them at risk of reduced accuracy when they do occur in evaluation data or find another way of training models for these rare monophone combinations, as discussed below. The results of experimenting with these two options are shown in Chapter 6.

5.3.1 Building Diphone Models from Well-trained Monophone Models

In the face of data scarcity, where there are insufficient training data to properly estimate all the model parameters, alternate measures are needed for the most affected classes. One possibility is to build the diphone models from monophone models that have been well-estimated from the same training data set. The much smaller size of the monophone set guarantees sufficient representation of all classes in the training set. At the very least, concatenated partial monophone models can be a very good starting point for unknown diphone models.

Figure 5.2 shows the construction of a diphone model from two 5-state left-to-right monophone models. The transition probabilities between the states are mostly kept intact, except at the beginning and end of the diphone model where the connections with the removed states were severed.

The HMM models used for the task of speech recognition in this thesis mostly have a left-to-right topology, which makes the process of isolating a segment within the model relatively simple. The two exceptions are the fully connected model used to model non-speech sounds and the single-state HMM used to model silence. Because these models are designed for continuous speech segments that do not carry a fixed temporal structure, the whole model is used as part of the diphone instead of just the “first or second half”.

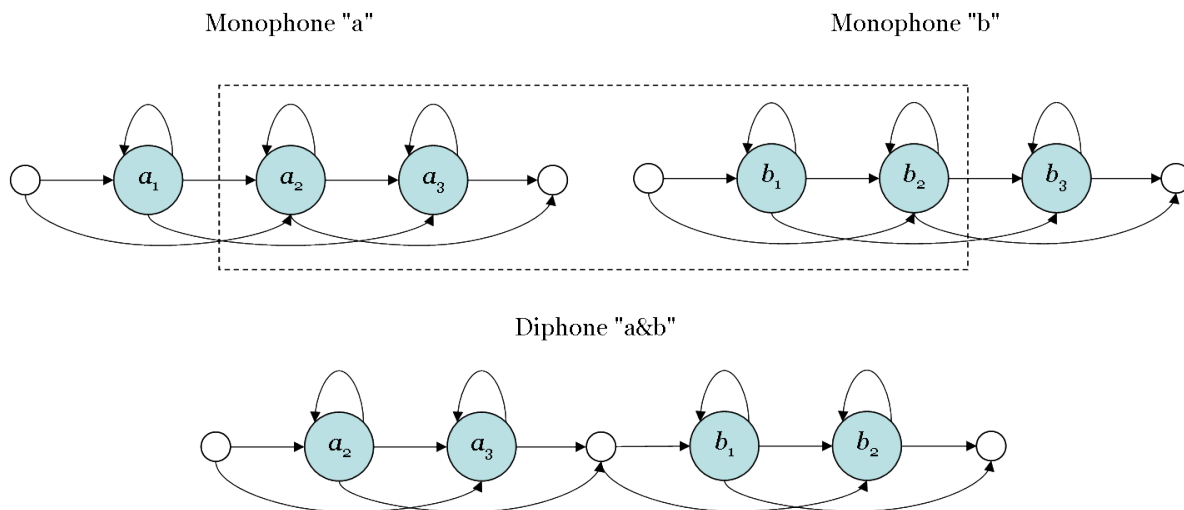


Figure 5.2: Construction of a diphone model from two well-estimated left-to-right monophone models.

5.3.2 Bootstrapping the Diphone Set with Monophone models

The construction of diphone models from monophone models as used in this research, was originally intended as a way of handling rare diphones not found in the training set. By constructing missing models from well-estimated monophone models, we can improve recognition accuracy where these diphones occur in the evaluation set. However, the entire diphone set can be bootstrapped by using monophone models to construct first approximation diphone models. These models can then be subjected to further training iterations to finely model the diphone characteristics and transitions, while retaining good estimations for rare diphones. This is done through a technique called *Maximum A Posteriori Estimation*, described in the next section.

5.4 Maximum A Posteriori Estimation

Maximum A Posteriori (MAP) estimation, also called Bayesian estimation or Bayesian learning, is used to incorporate prior knowledge into the training procedure by adding probabilistic constraints on the model parameters. It is commonly used to deal with the problems posed by data scarcity that would normally lead to inaccurate estimates of the model parameters. The MAP estimation procedure can be applied to two types of estimation problems – parameter smoothing and model adaptation, both resulting from the need for proper parameter estimation in the face of insufficient training data.

Parameter smoothing is used when common prior characteristics for all models can be assumed. This is followed by additional training on the limited model-specific training data available to fine-tune the unique model characteristics. Parameter smoothing is often used for applications such as speaker recognition to obtain speaker-specific models from more generic speaker-independent models where limited speaker-specific data is available. Model adaptation techniques use model-specific prior information obtained through other means (e.g. models reliably trained on large external data sets with similar conditions) and alter the model to conform to the more specific conditions of the current application. The MAP estimation used in this research is a type of model adaptation and is described below.

5.4.1 Mathematical Formulation

HMM parameters are usually estimated with the use of an expectation maximisation (EM) algorithm, such as the Baum-Welch algorithm described in Chapter 3, which applies a maximum likelihood (ML) estimation method to estimate the model parameters. The maximum likelihood estimation method can be adapted for use with MAP estimation by incorporating prior knowledge of the model parameters with a certain weight or importance associated with each of them.

Given a set of T training observations $\mathbf{x} = (x_1, \dots, x_t)$ to estimate the model parameter vector θ , the maximum likelihood estimate is calculated with the function

$$\theta_{MLE} = \arg \max_{\theta} f(\mathbf{x}|\theta) \quad (5.1)$$

where $f(\mathbf{x}|\theta)$ is the probability density function of the observation data given the model parameters.

The ML estimation calculations assume that the model parameters are fixed, but unknown beforehand. If an appropriate prior distribution of the model parameters to be estimated are available, $g(\theta)$, the parameters can be estimated using MAP estimation and the likelihood function changes to

$$\theta_{MAP} = \arg \max_{\theta} g(\theta|\mathbf{x}) \quad (5.2)$$

$$= \arg \max_{\theta} f(\mathbf{x}|\theta)g(\theta) \quad (5.3)$$

The choice of prior distribution family is important, because similarly to ML estimation, MAP estimation calculations are relatively easy when the distribution type possesses sufficient statistics of fixed dimension for the parameter vector θ . For mathematical stability the prior and posterior distributions should also belong to the same distribution

family for any sample observations. Therefore the prior distribution for the mean of a Gaussian density should also be a Gaussian density.

Hidden Markov Models contain concealed underlying parameters and as such complicate the MAP estimation calculations. Fortunately Dempster *et al.* [25] proved that an EM algorithm can be used to estimate model parameters by adapting the Baum-Welch algorithm to use MAP estimation where normal ML estimation was used before. The mathematical foundation for the definition of MAP estimation and its use in estimation of multivariate Gaussian mixture densities are explained in full in [35].

5.4.2 MAP Estimation of Gaussian Mean Values

The MAP estimation used in this research is limited to the estimation and adaptation of the Gaussian mean values of the output probability density functions (μ_j as defined in Equation 3.40). Other HMM parameters such as the starting probabilities (π), transition probabilities ($\mathbf{A} = [a_{ij}]$) and Gaussian variance of the output probability density functions (Σ_j) can also be adapted through the use of MAP estimation. These formulations will not be discussed here as the adaptation of the mean is sufficient to demonstrate the idea of MAP estimation for use in Hidden Markov Modelling.

The following formulation for the MAP estimate of the mean parameter for a Gaussian density is given in [66].

Let $\mathcal{N}(\mu, \sigma^2)$ be a component density in a Gaussian mixture density. If we assume the variance is known and fixed and the mean μ has prior distribution $P_0(\mu)$ with mean ρ and variance τ^2 respectively, the MAP estimate of the mean parameter from a set of N observations $\mathbf{x} = (x_1, \dots, x_n)$ is given by

$$\hat{\mu}_{MAP} = \frac{N\tau^2}{\sigma^2 + N\tau^2}\bar{\mathbf{x}} + \frac{\sigma^2}{\sigma^2 + N\tau^2}\rho \quad (5.4)$$

It can be seen from Equation 5.4 that if there are no training observations presented, $N = 0$ and the estimated mean is equal to the mean of the prior distribution, ρ . When training is done on a finite number of observations, the estimated mean becomes a weighted average of the prior mean ρ and the sample mean of the observations $\bar{\mathbf{x}}$. Ultimately, when infinite training observations are available, $N \rightarrow \infty$, and the best estimate of the mean parameter μ is equal to the the sample mean $\bar{\mathbf{x}}$. The variance of the prior distribution acts as a confidence measure between the prior estimate of the mean and the new ML estimate. As the variance of the prior mean approaches infinity ($\tau \rightarrow \infty$) the MAP estimate approaches the ML estimate, placing maximum confidence in the ML estimate of the observation data.

As the variance of the prior mean approaches zero ($\tau \rightarrow 0$) the confidence in the prior estimate is increased and trusted to be robust.

5.4.3 MAP Estimation as Used in this Thesis

In the research done for this thesis MAP estimation was used to partially handle data scarcity by improving parameter estimation. MAP adaptation is used to adapt prior estimates of the Gaussian mean values of the output probability density functions from well-trained monophone models with the limited labelled data available. As described in Section 5.3, the diphone models are built from monophone models that were trained on the same data set as the one used for model adaptation of the diphone models. The output probability density functions on the model states obtained from the well-estimated densities of the monophone models are used as prior estimates for the Gaussian mean values with a moderate confidence measure. The confidence measure ensures that enough training data must be available before the ML estimate of the observed data is trusted to be robust enough to outweigh the prior mean.

The effectiveness of the MAP adaptation technique can be increased by either utilising prior densities for more model parameters or by training the prior densities on large reliable external databases and subsequently increasing the confidence measure in the prior estimates. Studies have shown that recognition accuracy can be improved with the MAP estimation technique when compared to direct ML estimation, especially when available training data is limited [82].

The reason for using MAP estimation in this research only for the adaptation of the Gaussian mean values and not for the variances and other HMM parameters as well, is the need to keep the resource requirements of the speech recognition system relatively low. Each parameter with an additional prior estimate doubles its memory requirements which leads to a significant increase in the eventual model size, preventing additional MAP estimation training if insufficient system memory is available. The experiments described in this thesis had the specific objective to evaluate speech recognition systems based on different acoustic modelling techniques in low-resource environments – a realistic scenario for practical applications.

5.5 Decision Tree Based State Clustering

Larger model sets that incorporate more co-articulation information into the subword units on the same sized training set as used for monophone-based recognition systems, both increase resource requirements and lower parameter estimation reliability. There-

fore a reduction in the number of parameters to be estimated often leads to increased recognition accuracies due to the better utilisation of the limited training data. State clustering is an effective technique that can be used to group the output probability density functions (PDFs) of the HMM models, based on some similarity measure. Each PDF is associated with a state in an HMM model and when they are optimally combined with similar PDFs, either within the same model or across different models, a reduced set of classes are created with fewer parameters to be estimated. The labelled training data is pooled into the new clustered set of classes, increasing class representation and parameter reliability.

5.5.1 Overview of Decision Tree Logic

Decision trees are used to dynamically find a reduced set of classes based on a specific training data set and appropriate clustering parameters to establish control over the eventual size of the new class set [34]. The training data is grouped, based on the parameter attributes, to produce a reduced set of classes with maximised discrimination between them. The state clustering algorithm resembles a tree structure where

- the root node contains all possible datapoints,
- the branches of the tree represent the attribute values and decisions made to partition the data,
- the tree nodes represent the subsets of data groups and
- the leaf nodes represent the final set of class labels.

The input parameters can be either categorical, continuous or discrete. A multi-disciplinary survey and overview of research done on decision trees can be found in [56].

The decision tree is built using recursive partitioning at each tree node to optimally split the training datapoints. The optimal splitting criterion is the one that maximises the homogeneity in the subsequent partitions, which leads to better separation of the classes and aids the classification process.

To train a decision tree, both the tree structure and the splitting criterion used at each node in the tree have to be determined. The splitting criterion consists of the choice of input parameter and the corresponding value at which the datapoints on the node are subdivided. The tree is grown from one root node into a larger optimal tree by recursively adding nodes and splitting criteria from a set of possible candidates across all input parameters, until a convergence criterion is satisfied. Leaf nodes are reached

when all datapoints corresponding to the current node have the same classification. This method of decision tree building is called greedy top-down induction. The convergence criterion is usually a threshold based on a measure of improvement to system accuracy that can be attributed to the current changes.

As an alternative to the convergence criterion, the tree can be extended up to its limit, where each leaf node represents a single datapoint, and then pruned back to ensure that each leaf node represents more than a certain minimum number of datapoints. Pruning is done by collapsing internal nodes and combining the corresponding regions. Pruning the decision tree also improves overall performance, because the fewer datapoints a node is trained on, the less robust the resultant decisions can be, due to limitations and idiosyncrasies in the training data. Modelling decisions made on the basis of a few datapoints are often meaningless and sometimes harmful for classification if the rules are extended to larger data sets. The pruning step can be done on a development set of unseen data instead of the training data set to increase generalisation and avoid over-fitting on the training data.

A decision tree has the advantage that it is easily interpreted by humans because it involves a series of decisions with a finite number of outcomes, applied to the individual input variables. It is closely linked to our ability to explain a situation by first considering all options and then eliminating some possibilities on the basis of input observations.

Example

A good example of decision tree logic is found in [9]. Let's assume a single datapoint is a two-dimensional vector containing parameter values, which can be used to classify the datapoint as one of N possible classes. The input parameter space can be visualised as a two-dimensional Cartesian space with each axis representing the possible values for one input parameter. The input space will ultimately be divided into N regions for which unknown datapoints that fall in these regions will be classified accordingly. The class boundaries are determined by the chosen splitting criteria. For example, if the input parameter x_1 is smaller than the value θ_1 , it falls in a different group than if it were larger, therefore $x_1 > \theta_1$ is defined as the first splitting criterion. The two-dimensional input space of Figure 5.3 is recursively segmented by first creating two subregions according to model parameter θ_1 and then independently subdividing these regions further according to other model parameter splitting criteria. The corresponding decision tree, also shown in Figure 5.3, reflect the decisions made when classifying unseen input data.

At the root node all datapoints are grouped together. The first splitting criterion evaluates parameter x_1 of the input data around value θ_1 , subdividing the datapoints

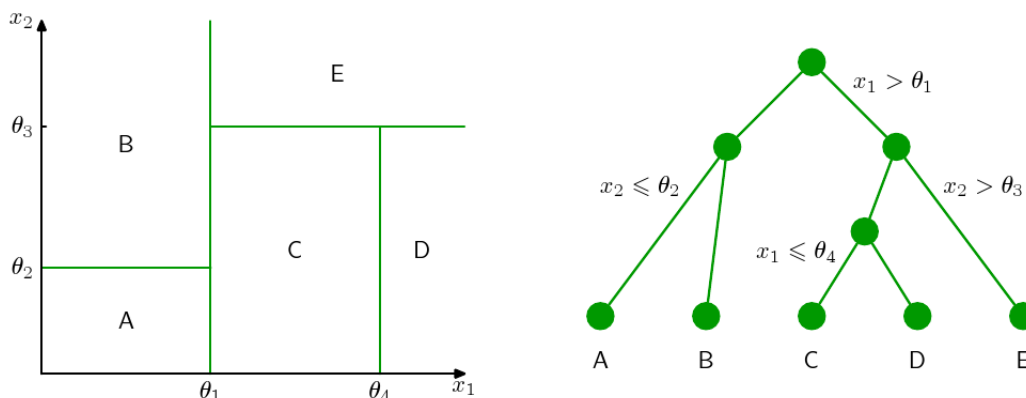


Figure 5.3: A partitioned two-dimensional input space that has been divided into five regions using axis-aligned boundaries and the corresponding binary decision tree.

into two groups. For the first group where $x_1 \leq \theta_1$, parameter x_2 of the input data is evaluated around value θ_2 , subdividing the datapoints further into two groups which can be classified as either class A or B. Similarly the data in the group where $x_1 > \theta_1$ is recursively subdivided to yield three groups that can be classified as either class C, D or E. Classification of an unseen datapoint is done by traversing the tree from the root node, evaluating the input parameters according to the attribute values and partitioning decisions defined for each node, and classifying the datapoint according to the class label of the resulting leaf node.

5.5.2 Classification and Regression Trees (CART)

Classification and Regression trees (CART trees) are decision trees that can be used for either classification or regression problems. The use of CART algorithms in statistics and pattern recognition was popularised in 1984 by Breiman *et al.* [13] and are now extensively used in the fields of statistics, pattern recognition, decision theory, signal processing and machine learning. The classification problem consists of labelling an input datapoint as belonging to one of a set number of classes by evaluating its parameters. An example of the classification problem is classifying a tree as part of a botanical family based on its physical parameters, such as leaf size and colour. The regression problem consists of reducing an input datapoint with more than one parameter to a single numerical output variable, which approximates a real-valued function. An example of the regression problem is predicting the selling price of a single family home from a set of parameters such

as the number of bedrooms and whether or not the house has a pool. The decision tree methodology for these two problems are similar, with the only differences found in the mathematical evaluation of splitting criteria, convergence thresholds and system performance. The application of CART trees in this thesis only pertains to a special case of the classification problem – reducing the size of a set of possible classes while preserving the essential characteristics to provide an accurate summary – and as such the discussion of CART trees will be limited to the methodological and mathematical considerations of using CART trees for classification. The background theory and implementation considerations discussed here are based on several sources [9, 23, 29, 50, 84, 43].

5.5.3 Creating a CART

The objective of using CART trees in this research is to find an optimal set of possible output classes into which the training data can be partitioned in such a way as to maximise discrimination between the classes and homogeneity within the classes. The splitting criterion with the highest probability of achieving good data partitions reveal the input parameter that holds the critical data at this stage in the partitioning process. The CART tree induction process is therefore data-driven and the optimal partitioning of the training data into a new set of output classes is done with little dependence on the use of external knowledge.

In order to build a CART tree we need the following:

1. an accurately labelled training set,
2. a set of possible splitting criteria or questions to be considered at each internal node in the tree,
3. a way of defining the optimal splitting criterion to use at each step in the partitioning process,
4. a strategy to decide when the optimal tree size has been reached, indicating the extent of tree induction and pruning and
5. a way of determining the class-tag to be associated with each leaf node.

Labelled Data set

The original set of possible classes consists of the output probability density functions found on the states of all the subword model HMMs. The training data is labelled according to these original class labels and is therefore extremely sensitive to the proper

alignment and correct labelling of the data. A small realignment of class boundaries in a training file can have a significant impact on the structure and outcome of the resulting CART tree. Because of this it is necessary to dynamically align the training set in order to find an accurate start and end time, and therefore the set of relevant feature vectors, for each class label.

Question Set

The question set is used to provide the CART system with a set of logical splitting criteria with three broad categories:

- incorporation of linguistic knowledge into the clustering procedure by providing logical groupings of phones that are linguistically similar,
- external knowledge pertaining to the structure of the relevant subword models and the nature of the required clustering procedure and
- questions regarding the discrete value of a single variable as one of a set of predefined classes, such as phones, called *singleton* questions.

The question set used for diphone models in this research focuses on either the first half or second half of the diphone model or the state number within these models. The assumption is made that output probability densities on HMM states representing linguistically similar diphone characteristics will tend to be similar enough to be grouped together. Examples include: “*Is the diphone transitioning from a fricative?*”, “*Is the diphone transitioning to the vowel /æ/?*” and “*Is this output class from the first state of an HMM model?*”.

The question set described above create binary CART trees, where each internal node in the tree can only have two children nodes. This is due to the fact that the questions are structured to provide yes/no answers. The binary CART tree provides more stability and scalability for its intended use in this thesis, which is controlled clustering of individual HMM states. Aggressive splitting during top-down construction of the tree, as is the case when the data on a node is split into more than two groups, will result in ill-conditioned CART trees.

Splitting Criteria

In order to decide on the best possible split of the training data at a certain internal tree node, we need a measure of the difference each possible split, or question, will make to the overall performance of the tree. In most applications the gain is measured in terms of the

proportion of misclassified datapoints or the variance associated with misclassification. A variance can only be calculated when there is a certain degree of misclassification that can be associated with each datapoint or in cases where misclassification of a specific type of datapoint is more catastrophic than others. The calculated cost of misclassification for all training datapoints must be minimised at each node split in order to maximise the overall gain in classification accuracy. These calculations are done for each possible question in the question set to determine the optimal question to use as splitting criterion at the current node.

Tree Induction

The induction of the CART tree is usually done by some measure of diversity such as entropy or the Gini index [13]. The pruning, on the other hand, is done by estimating the misclassification error via cross-validation. In contrast with determining the optimal tree size in two steps, rules governing induction and pruning of the tree can be done from a single criterion, the *Minimum Description Length* (MDL) principle, described in Appendix C.

Leaf Node Classification

The class-tag assigned to each leaf node is equal to the most frequently occurring class label associated with the training data on the leaf node.

5.5.4 Classification and Regression Trees as Used in this Thesis

Decision tree-based state clustering is used to reduce the total number of parameters to be estimated in the diphone-based phoneme recognition system in such a way as to maximise the recognition accuracy in a low resource environment. The new set of possible output classes is the optimal grouping of subsets of the original class set, resulting in clustering of output probability density functions on the HMM states, or state clustering. The clustering procedure exploits the similar effect of certain contexts on the sound being modelled. The use of CART trees enable better utilisation of limited training data when a large number of classes are to be estimated by pooling their respective datapoints, leading to better estimated models. It also significantly reduces the memory requirements and computational load necessary when using the models for speech recognition.

The research done for this thesis used an existing implementation of the CART algorithm in the PatrecII software developed at the University of Stellenbosch. A labelled data set and clustering parameters are used as input to obtain a map file that maps each

original class label to a new cluster label. This map file is then used to transform the labelled data set from which new hidden Markov models are trained. The question set used for diphone and biphone-based systems is adapted from an existing question set designed for triphone models and can be found in Appendix C.

The advantage of using CART trees for state clustering is the simplicity and scalability of this technique. The data-driven character of CART trees ensures that general trends in observation data with regards to the outcome is automatically identified, revealing more about the input data than could be assumed with *a priori* information. CART trees are easy to implement and use, and the results are easily interpreted and understood. The clustering of output probability densities of HMM states are done in such a way that the final number of classes are scalable. Leaf nodes can be easily combined into their parent nodes to reduce the total number of leaf nodes if necessary.

The use of CART trees for model state clustering has the added benefit of providing a way to synthesize models that do not occur in the training data set, but may occur during evaluation of unseen data. The tree structure allows for easy regression to higher, more general levels within the tree for any unseen context or class with ill-conditioned parameter estimation caused by data scarcity. Inter-word context dependencies can also be handled through regression modelling from CART trees to aid in lexical decoding [22].

5.6 Summary

In this chapter, various adaptation techniques that were used to enhance the diphone-based recognition system are explained.

A splitting technique is used to reduce the total number of basic phonemes from which the diphones are defined by splitting “double” phonemes, such as diphthongs and affricates, which characterise the transitions between two phonemes that are already in the data set. Because diphones inherently model transitions, these phonemes are left out of the original set of phonemes used, thereby creating a much smaller set of diphone models without losing any modelling capability.

During decoding, a basic diphone grammar is used to restrict diphone transitions to only fit diphones together that model two halves of the same phoneme. The diphone grammar is still classified as a 0-gram grammar, as all valid diphone transitions have equal probability of occurring.

Diphone set completion is used to create unseen diphones that were not found in the training set by building them from well-estimated monophone models. This procedure was eventually used to build all diphone models from well-estimated monophone models that

could be used as prior estimations for further MAP estimation training. The mathematical formulation and theoretical background of MAP estimation are provided, together with a description of how MAP estimation was used in this research. The MAP procedure was only used on the Gaussian mean values of the output probability density functions due to the requirement for implementation in a low-resource environment.

Decision-tree based state clustering is used to reduce the total number of model parameters through state-level tying. The background theory of decision trees, and more specifically classification and regression trees (CART) are discussed, together with important considerations such as CART training, the question set used, splitting criteria, tree induction and leaf node classification. CART trees were used in this thesis for state-level clustering of output probability distributions to drastically reduce the total number of model parameters, increasing the class representation in the data set and subsequently increasing recognition accuracy. Various experiments on the use of CART trees in diphone-based recognition systems can be found in Chapter 6.

Chapter 6

Experimental Investigation

The main purpose of this research is the evaluation of diphones as basic units for acoustic modelling. This evaluation was done by direct comparison of a baseline monophone system with intermediate diphone adaptations. The best diphone configuration was ultimately compared to context-dependent biphone acoustic models in similar parametric conditions.

6.1 Experimental Setup

6.1.1 Hardware Platform

All experiments described in this chapter were run on the same computer and under the same conditions to enable direct comparison of the resulting execution times. The computer has an Intel dual core 1.8GHz processor with 4GB of system memory. This hardware platform configuration was chosen to simulate operating conditions for typical speech recognition applications. Running the experiments on a distributed system or one with more than a standard number of processors would significantly reduce the execution time of individual experiments, but these systems do not constitute low-resource environments. Also, the shorter the execution time of each experiment, the harder it is to objectively compare the results. Being able to run a speech recognition engine on a standard desktop computer has many advantages and is of practical use.

6.1.2 Software Platform

All algorithms used in this research were implemented in C++ within the framework of the PatrecII software suite. PatrecII is a large collection of C and C++ code designed and implemented by various members of the Digital Signal Processing group within the Department of Electrical and Electronic Engineering of the University of Stellenbosch.

The PatrecII software suite contains modules typically needed for speech processing research. Additions were made to the system where needed, especially for the handling of diphones.

6.1.3 The AST Data set

All systems were trained and evaluated using the English subset of the African Speech Technology (AST) Data set [51, 59]. The speech database was collected over both mobile and fixed telephone networks and manually transcribed – orthographically and phonetically – for use in speaker-independent continuous-speech recognition systems. The English database consists of contributions by both mother-tongue and non-mother-tongue speakers and can be divided into five equally sized categories as spoken by English (EE), Afrikaans (AE), Black (BE), Coloured (CE) and Asian (IE) participants. In order to make provision for highly variable speech amongst non-mother-tongue speakers, all these sets are incorporated into the English database. However, for the purpose of this research, the BE subset was omitted because the phonetic changes in word-pronunciation in this subset are too far outside the norm when compared to the other subsets. If included this would have led to a significant drop in system accuracy. Each subset consists of recordings of between 300 and 400 different speakers, each of which contributed 40 utterances of a mixture of spontaneous and read speech.

The duration statistics for the data sets used for all experiments in this research is shown in Table 6.1.

Table 6.1: *Duration statistics for the training and testing data sets from the English subset of the AST speech database.*

Data set	Total speech file duration	Average speech file duration
Training	116686 sec (32h 24m 46s)	11.2991 sec
Testing	12655.4 sec (03h 30m 55.4s)	11.2994 sec

For more information on the AST data set, please refer to Appendix B.

6.1.4 Signal Processing

The speech signal waveforms are obtained at a sampling rate of 8kHz and encoded using a-law compression. Feature extraction is done by calculating 12-th order Mel frequency Cepstral Coefficients (MFCCs) from overlapping 30 msec Hamming windowed portions of the signal, centered every 15 msec. These values are then stored in feature files.

A set of mathematical functions (a feature normaliser) is used to collectively transform the feature vectors, based on the set of possible classifications to be employed in the recognition task. Normalising the input data set is mathematically equivalent to finding a perspective on the data that will improve class discrimination. The normaliser used for all systems described in this research contains the following components:

- **Cepstral mean subtraction:** used to shift features on a per utterance basis to have a zero mean.
- **Scaler:** scales data by dividing by a scaling vector (special case of linear projection) for each dimension on the feature vector to have unity variance across the whole training set.
- **Incorporation of context:** divides the features into overlapping frames, combining the features inside the frame to create one “super” feature vector. A framelength of 9 is used, each time shifting by one feature.
- **Linear Discriminant Analysis (LDA):** used to reduce the feature dimension from 108 to 24, by means of a class-based Karhunen-Loève Transform (KLT).

The normaliser is trained on the training data set and subsequently used to transform any input feature data before being used in the recognition system.

6.1.5 Statistical Modelling Parameters

All acoustic models are left-to-right hidden Markov models with tree-based diagonal-covariance Gaussian mixtures as output probability density functions. Diagonal-covariance Gaussians were used because of their efficient use of resources at a minimal cost in reduced accuracy. The Gaussian mixture trees are grown to a maximum of 256 leaf nodes with a top-down tree-building algorithm through iterative training – each tree node is split into further mixtures only if enough training data is available for accurate parameter estimation.

The training of hidden Markov models occurs in two phases. The output probability densities are first pretrained from the labelled data set using a specification file containing a list of class labels and a probability density function (PDF) prototype for each class. Each PDF starts as a single representative Gaussian density, which is then iteratively trained on the labelled training data to create a tree-based Gaussian mixture density. The second training phase occurs when these pretrained densities are combined with HMM prototypes, one for each subword unit, to create a set of initial models. The mapping of probability density functions to HMM states is described in an HMM specification

file. Embedded re-estimation training is used to iteratively train all the HMM parameters on the same training set as the one used for pretraining the probability densities. The re-estimation procedure utilises a Viterbi segmenter and Baum-Welch re-estimation algorithms.

6.1.6 System Evaluation

For system evaluation the hidden Markov models are combined into a “super” HMM structure according to the grammatical structure imposed by grammatical modelling to form a decoder model. Null states are used to connect the individual subword HMMs into a network structure where the link probabilities represent the probability of transitioning from one model to the next. The decoder used for the diphone-based system (shown in Figure 5.1) differs significantly from the monophone-based system (shown in Figure 3.6).

The decoder is then used by a Viterbi segmenter to create new transcription files for the testing data set. In the diphone-based system the transcriptions are relabelled to convert diphone sequences to phoneme sequences, with the same applying to biphone-based systems. To evaluate system performance, the decoded phoneme sequence is compared to the manually transcribed reference sequence by means of a dynamic time-warping algorithm as described in Section 3.3. In addition to noting the execution time needed to decode the testing data set and the recognition accuracy, it is necessary to know whether the relative improvement of one system over another is statistically significant or merely by chance. To do this we used the matched-pairs test described in the next section.

6.1.7 Statistical Significance Tests

The matched-pairs test was used as described in [37, 63]. It is a statistical significance test often used to evaluate the differences in test results of two continuous speech recognition systems, because it does not require the errors within a segment to be statistically independent.

The test set is divided into segments in such a way that the errors in one segment is statistically independent of the errors in another segment, such as phrases or sentences. The AST testing data set contains a set of 901 files recorded separately, each containing speech utterances of differing length. These utterances are therefore statistically independent and comply with the conditions required by the matched-pairs test.

Let A_1 and A_2 be the two algorithms to be compared, N_1^i the number of errors made on the i -th segment by A_1 and N_2^i the number of errors made on the i -th segment by A_2 . The type of error is unimportant as long as the same measure is used for both algorithms. For

this research the error count is defined as the total number of substitutions, insertions and deletions found in a speech segment. For each segment, the error difference is calculated

$$Z^i = N_1^i - N_2^i \quad i = 1, 2, \dots, n \quad (6.1)$$

where n is the total number of segments. If μ_Z is the unknown average difference in the number of errors in a segment made by the two algorithms, we would like to ascertain whether $\mu_Z = 0$. The natural estimate of μ_Z is

$$\hat{\mu}_Z = \frac{1}{n} \sum_{i=1}^n Z_i \quad (6.2)$$

and the estimate of the variance of Z_i is

$$\hat{\sigma}_Z^2 = \frac{1}{n-1} \sum_{i=1}^n (Z_i - \hat{\mu}_Z)^2 \quad (6.3)$$

If n is large enough and $\mu_Z = 0$, the distribution of the statistic function

$$W = \frac{\hat{\mu}_Z}{\left(\sqrt{\frac{\hat{\sigma}_Z^2}{n}}\right)} \quad (6.4)$$

will approximately be a Normal distribution with zero mean and unit variance, $\mathcal{N}(0, 1)$. To test whether $\mu_Z = 0$, we compute

$$P = 2Pr(Z \geq |\omega|) \quad (6.5)$$

where Z is a random variable with Normal distribution having a 0 mean and unit variance, and ω is the realised value of W . If $P < \alpha$ for some significance level α , the assumption that $\mu_Z = 0$ cannot be true and the difference between the two tested algorithms are statistically significant within an error margin of α . Values for α are typically 0.05, 0.02 or 0.001.

Only recognition systems that utilise the same phoneme class set can be directly compared. In this research the monophone baseline system, intermediate diphone adaptations and optimised biphone-based recognition systems are all tested in pairs, where the test was deemed appropriate, to determine the statistical significance of their phoneme recognition results.

6.2 Monophone-based Continuous Phoneme Recognition

6.2.1 Motivation

For objective analysis of the diphone models, a monophone-based continuous phoneme recognition system was built to use as a baseline for subsequent evaluation experiments.

The baseline system represents a standard implementation of a context-independent system, which is commonly used for fast, simple acoustic modelling. Using a baseline system isolates the measured improvement in system accuracy due to the choice of subword unit from the influence of external factors, such as the speech corpus, type of speech recognition, signal handling, feature vector calculation and mathematical modelling algorithms. The monophone-based system provides standards for both system accuracy and execution time and subsequent experiments are evaluated according to these values.

6.2.2 Experimental Setup

Signal handling and statistical algorithms and parameters used for the monophone-based system are all exactly as described in Section 6.1.

The monophone model set consists of the 44 phonemes described in Table A.1 including a model for silence (SIL) and a model representing non-speech sounds (OTHER).

The data set contains accurate phonetic boundaries, with each phoneme segment equally divided into the number of HMM states for that phoneme model. These sub-segment boundaries are considered good first approximations and will be adjusted during iterative training.

The output probability densities are tree-based diagonal-covariance Gaussian mixtures with a maximum size of 256 for phoneme states, 64 for SIL states and 8 for OTHER states.

To improve accuracy the longer duration phonemes, such as diphthongs and affricates, are modelled with four states, whereas shorter sounds are sufficiently modelled with three – the transition from previous sound, the steady-state portion and a transition to the next sound. The SIL model is represented by a one-state HMM due to its constant deterministic nature, and the OTHER model is represented by a three-state fully-connected HMM due to its random non-deterministic nature.

The monophone decoder used is depicted in Figure 3.6 and utilises a 0-gram grammar. Each monophone therefore has an equal probability of occurring next. System evaluation is done through measuring the number of phoneme errors in terms of substitutions, deletions and insertions, as described in Section 3.4.

6.2.3 Results

Accuracy

The monophone recognition system achieved a recognition accuracy of 46.1%. The breakdown of evaluation results are shown in Table 6.2.

Table 6.2: *Continuous Recognition Accuracy: Monophone Baseline System*

Correct	Substitutions	Deletions	Insertions	Accuracy
54.10%	32.83%	13.07%	08.43%	46.10%

Execution Time

The monophone-based system contains a total of 139 output probability density functions. Decoding of the testing data set was done in **9 min 14.5 sec**.

6.2.4 Interpretation

There are a few techniques that could slightly increase the recognition accuracy of the monophone-based system, such as using full-covariance Gaussian densities, daisy-chaining the training process to incorporate interim realignment of the original training set and using a lexicon to do simultaneous alignment of multiple segmentation levels for better re-estimation. These techniques were not applied to the diphone-based systems for a number of reasons (limitations on resources and no available diphone-based lexicon) and were therefore omitted in the final monophone experiments to be able to directly compare the diphone results to the baseline system.

The main advantage of the monophone-based system is the speed of execution. The testing data set, which is approximately 3 and a half hours of speech data, was decoded in 10 minutes. This means the speech recognition system works approximately 22 times faster than real-time, making it a very efficient recognition system.

6.3 Diphone-based Continuous Phoneme Recognition

In the analysis of diphone-based continuous phoneme recognition systems, this set of experiments was designed to isolate contributions made by each adaptation technique for better diphone modelling. A tree-structured diagram shown in Figure 6.1 reveals the relationship between these experiments.

Signal handling and statistical algorithms and parameters used for the diphone-based system are all exactly as described in Section 6.1.

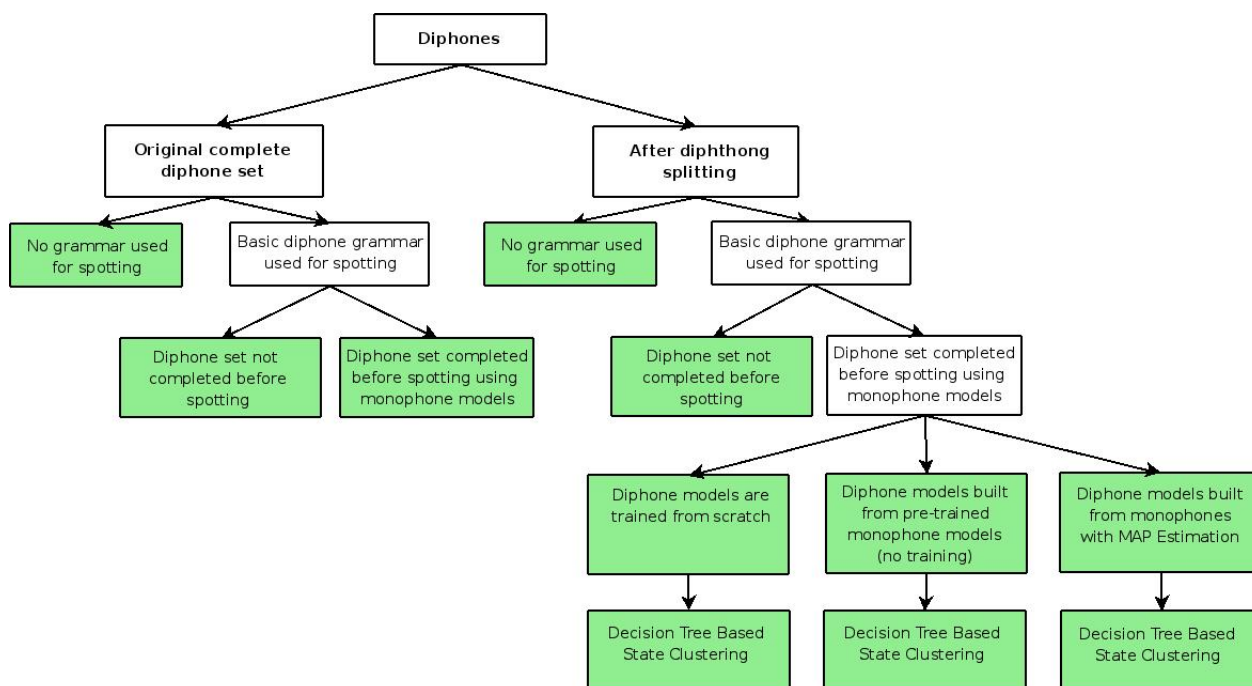


Figure 6.1: Summary of diphone-based experiments designed to isolate contributions by various adaptation techniques. Coloured boxes represent unique configurations of the diphone-based system for which experiments were run. Each branch represents a system design decision made, each of which can be attributed to an adaptation technique as described in Chapter 5

6.3.1 First Approximations

Motivation

The first set of experiments were used to evaluate the use of diphones as subword units if they were treated in the same way as monophones. These experiments were done to identify a large percentage of the implementation issues associated with converting a monophone-based system to one using diphones. The feasibility of this project depended on the ability to incorporate the use of diphone-based systems in the PatrecII software suite. Addressing implementation issues early with a basic “first approximation” system provided an excellent framework from which to improve the diphone-based recognition system.

One issue encountered during this experiment is ineffective decoding due to the exponential increase in the number of models from the monophones to a diphone-based system. If all models have an equal probability of occurring, the margin of error increases with the model set size. Fortunately, diphone models are by their very nature constrained to a basic diphone grammar – only diphones that start with the last half of a specific phoneme can follow a diphone model that ended with the first half of that phoneme. This relationship between the diphone models can be used to create a specialised diphone decoder to aid in accurate phoneme recognition.

Experimental Setup

The data set was created from the phonetically aligned monophone data set described in Section 6.2. Diphones are defined as spanning from the middle of one phoneme to the middle of the next, and the diphone boundaries were calculated accordingly. Each diphone segment is then equally divided into the number of HMM states associated with that diphone model. Placing diphone boundaries in the centre of a phoneme segment is considered a good first approximation. Although the central frame may not be the actual midpoint of the sound, the boundaries are in a stationary portion of the sound, which make them more robust to small changes in segmentation so that the effect of this is not significant. Each utterance is assumed to start and end with silence, therefore the monophone model SIL is used as a “half-diphone” to be able to create the first and last transition of the utterance.

Using the monophone set from Section 6.2, there are 1595 diphones defined in the training set and 12 diphones were defined in the testing set that do not occur in the training set.

The output probability densities are tree-based diagonal-covariance Gaussian mixtures

with a maximum size of 256 for all diphone states, including the HMM states representing sections of the monophones SIL and OTHER.

The diphone models are all 3-state left-to-right HMMs with one skip-forward link at each state. Although this configuration is not ideal for modelling diphones, it was used as a simple first approximation. As the data scarcity problem is not addressed at this stage, increasing the total number of emitting states will only worsen the reliability of estimated parameters. All training and estimation procedures are exactly as with monophone models.

Three separate experiments were done on the training set of first approximation diphones:

1. **No decoding grammar with incomplete diphone set**

The 0-gram decoder depicted in Figure 3.6 was used to decode diphone sequences. Each diphone model therefore has equal probability of occurring next. Only diphone classes found in the training set were trained and used, with no attempt at diphone set completion or handling of unseen contexts.

2. **Basic diphone grammar with incomplete diphone set**

The diphone decoder depicted in Figure 5.1 was used to decode diphone sequences. The next diphone in the sequence is constrained to represent a transition starting from a specific monophone, reducing the number of possibilities to at most the number of monophones. Each of these subsets of possibilities have equal probability of occurring, thereby remaining a 0-gram grammar. To isolate the effect of the diphone spotter no diphone set completion was attempted. Only diphone classes found in the training set were trained and used, with no attempt at diphone set completion or handling of unseen contexts.

3. **Basic diphone grammar with diphone set completion**

To isolate the influence of diphone set completion, the previous experiment was repeated after diphone set completion. Models that are found in the testing set but not in the training set were constructed from monophone models and included in the diphone set used for decoding.

Results

The breakdown of evaluation results of the three experiments described above are summarised in Table 6.3.

The first approximation diphone-based system has significantly more densities than the monophone-based system and it takes considerably longer to decode the testing data

Table 6.3: *Continuous Recognition Accuracy: First Approximation Diphone System*

Experiment	Correct	Subs.	Deletions	Insertions	Accuracy
No grammar; Incomplete set	31.24%	63.87%	04.90%	15.91%	15.91%
Diphone grammar; Incomplete set	68.359%	28.657%	02.983%	23.970%	37.772%
Diphone grammar; Complete set	68.358%	28.658%	02.983%	23.970%	37.771%

set. The results are shown in Table 6.4.

Table 6.4: *Decoding Execution Time: First Approximation Diphone System (Execution Time is depicted as hh:mm:ss.ms) with additional value in terms of real-time (RT)*

Experiment	Number of Densities	Execution Time
No grammar; Incomplete set	4785	01:18:41.31 (Approx. 2.7 times faster than RT)
Diphone grammar; Incomplete set	4785	01:18:28.56 (Approx. 2.7 times faster than RT)
Diphone grammar; Complete set	4924	01:24:15.70 (Approx. 2.5 times faster than RT)

Interpretation

The first approximation diphone-based system is far less practical than the monophone-based system. The significant increase in number of densities (3442%) causes both lower recognition accuracy due to poor parameter estimation and a higher execution time (2.68 times faster than real-time) due to increased resources and decoder complexity.

The use of a basic diphone grammar for decoding provides the first meaningful improvement in system accuracy, although the system is still less accurate than the monophone-based system and far less efficient (2.69 times faster than real-time).

Completing the diphone set before decoding has little noticeable effect on system accuracy. This is due to the fact that the diphones that had to be built from monophones

to complete the set are only 0.75% of the complete set and are able to influence only a very small number of the calculations. The recognition accuracy is actually lowered slightly because of the additional number of models in the diphone set and the subsequent increase in decoder complexity.

6.3.2 Diphthong Splitting

Motivation

Diphthong splitting exploits characteristics inherent in diphones to drastically reduce the number of models, which directly addresses one of the major drawbacks of using diphones as subword units. These reductions have a significant effect on the recognition system, because fewer models lead to better estimated parameters and reduced computational requirements. And because these improvements come at no cost to the recognition system, diphthong splitting is an attractive adaptation technique to use early on.

Experimental Setup

The data set was created by first converting the phonetically aligned monophone data set described in Section 6.2 to utilise the smaller phoneme set. By splitting each diphthong or affricate phoneme into its constituent monophones (for example /*av*/ is converted to the sequence of monophones /*a*/ and /*v*/), the monophone set is reduced from 44 to 34 monophones in total. The data set is relabelled by inserting a new phoneme boundary in the centre of the diphthong segment. Similar to the data set for the experiment done in Section 6.3.1, the diphone segments span from the middle of one phoneme to the middle of the next. Each diphone segment is then equally divided into the number of HMM states for that diphone model.

Using the reduced monophone set, there are 997 diphones defined in the training set and 7 diphones are found in the testing set that do not occur in the training set. Conversely, after diphthong splitting the diphone set is 63.3% of its original size.

The output probability densities are tree-based diagonal-covariance Gaussian mixtures with a maximum size of 256 for all diphone states, including the HMM states representing sections of the monophones SIL and OTHER.

The diphone models are all 3-state left-to-right HMMs with one skip-forward link at each state, exactly as in the experiment done in Section 6.3.1. All training and estimation procedures are exactly as with monophone models.

Three separate experiments, similar to those described in Section 6.3.1, were done on the training set for diphones after diphthong splitting:

1. No decoding grammar with incomplete diphone set

The 0-gram decoder depicted in Figure 3.6 was used to decode diphone sequences.

2. Basic diphone grammar with incomplete diphone set

The diphone decoder depicted in Figure 5.1 was used to decode diphone sequences. Only diphone classes found in the training set were trained and used, with no attempt at diphone set completion or handling of unseen contexts.

3. Basic diphone grammar with diphone set completion

To isolate the influence of diphone set completion, the previous experiment was repeated after diphone set completion. Models that are found in the testing set but not in the training set were constructed from monophone models and included in the diphone set used for decoding.

Results

The breakdown of evaluation results of the three experiments described above are summarised in Table 6.5.

Table 6.5: *Continuous Recognition Accuracy: Diphone System after Diphthong Splitting*

Experiment	Correct	Subs.	Deletions	Insertions	Accuracy
No grammar; Incomplete set	33.31%	60.07%	06.62%	12.17%	20.37%
Diphone grammar; Incomplete set	68.868%	27.179%	03.952%	20.238%	44.498%
Diphone grammar; Complete set	68.867%	27.181%	03.951%	20.236%	44.500%

The number of densities and the decoding times for each experiment are shown in Table 6.6.

Interpretation

Using the diphone characteristics to facilitate diphthong splitting yields significantly improved results and a system that approaches feasibility. The system has an average increase in recognition accuracy of 7% across all three experiments, due solely to the reduction in the number of models.

The use of a basic diphone grammar once again provides a meaningful improvement in system accuracy, although the system is still less accurate than the monophone-based

Table 6.6: *Decoding Execution Time: Diphone System after diphthong splitting*
(Execution Time is depicted as hh:mm:ss.ms) with additional value in terms of real-time
(RT)

Experiment	Number of Densities	Execution Time
No grammar; Incomplete set	2991	00:56:54.77 (Approx. 3.7 times faster than RT)
Diphone grammar; Incomplete set	2991	00:56:47.13 (Approx. 3.8 times faster than RT)
Diphone grammar; Complete set	3130	01:01:04.05 (Approx. 3.5 times faster than RT)

system. Completing the diphone set before decoding again has little noticeable effect on system accuracy (the percentage of diphones built from monophones that were added is now 0.70% of the complete set, using the reduced monophone set) for the same reasons as described in Section 6.3.1.

All further diphone experiments will use the reduced monophone set as a base and a basic diphone grammar for decoding.

6.3.3 MAP Estimation

Motivation

Building diphone models from monophone models is a very effective way of dealing with data scarcity. By not relying solely on sufficient class representation, model parameters are well-estimated on the same training set than would originally be used. However, using these monophone-built models to decode would be impractical because at most it can yield exactly the same system accuracy as the monophone-based system, but will take much longer to do so because of the increased resource requirements. But using the monophone-built models as a starting point for further parameter estimation will solve both the data scarcity problem and effectively model diphone characteristics from the data set.

Experimental Setup

Firstly the diphone models are created as described in Section 5.3 using the monophones trained in the monophone experiment in Section 6.2. The list of diphones created in this way contains all diphone models found in both training and testing data sets based on the reduced monophone set obtained after diphthong splitting. The HMM models now contain a variable number of states, depending on the monophone models they were built from. Most monophone models contain 3 states and when two of these are combined into diphone models the resulting HMM has 4 states. The monophones SIL and OTHER are special cases and not “cut in half” when used to build diphones, instead they are used in their entirety.

These monophone-built diphone models are evaluated using the diphone decoder depicted in Figure 5.1, but as shown in Figure 6.2, duplication of HMM states occur in the spotter due to the way they were built from monophones.

The duplicate states were removed through a process of model pruning. By inserting additional skiplinks with appropriate link weights, the duplicate state can be removed and the original monophone state sequence retrieved. An example of this pruning method is shown in Figure 6.3.

After pruning, the spotter is the equivalent of the original monophone spotter and as such can generate identical monophone sequences. The pruning algorithm was only used in this special case of decoding diphones directly built from monophones.

The output probability density functions attached to the HMM states are now used as prior models for MAP estimation training. The prior model represents a certain amount of “previously seen” training data with an associated weight signifying the amount of previously seen data. If the prior model has a large weight relative to the number of new training examples, the new model will assume that the prior model is well trained and will avoid poor parameter estimation by keeping the deviation from the prior model to a minimum. MAP estimation gives the diphones the best possible starting point from which to be re-estimated and is done on the same training set that is used to train the monophone models, only now the data is diphone-labelled.

Using the reduced monophone set, there are 1004 diphones defined in both the training and testing sets. The output probability densities are cloned from monophone models and are therefore tree-based diagonal-covariance Gaussian mixtures with a maximum size equal to their equivalent densities in the monophones.

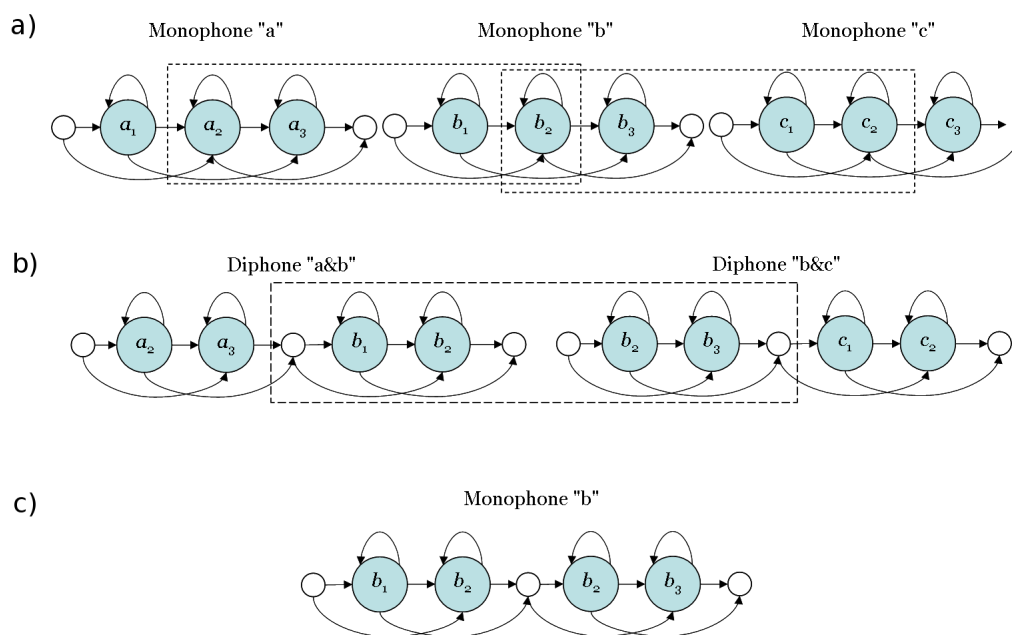


Figure 6.2: *Diphones built from monophone models and stitched back together in the diphone spotter. Subfigure a) shows a sequence of three phonemes and the states that will be used to build two diphone models corresponding to the transitions between the phonemes. Subfigure b) shows the constructed diphone models next to each other, clearly showing the duplicated state b_2 . When these two diphone models are linked in the diphone spotter structure, the configuration shown in subfigure c) occurs where the new “monophone b” is not equivalent to the original “monophone b”.*

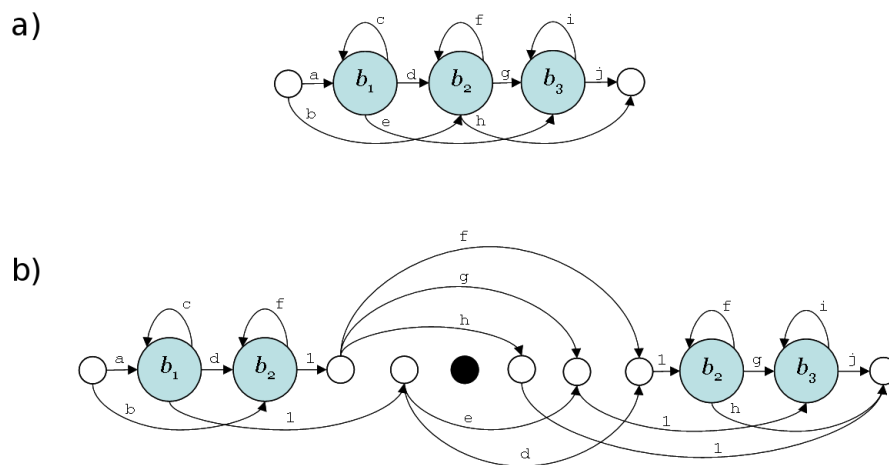


Figure 6.3: *Inserting skiplinks between two diphone models in the spotter structure to remove the duplicate state. The original monophone model is shown in a). The two halves of the diphone models built from the original monophone are connected in such a way as to create an HMM model equivalent to the original monophone model by adding skiplinks between them as shown in b). The null state coloured black would have been the original null state connecting the two diphone models.*

Results

The breakdown of evaluation results of the two experiments described above are summarised in Table 6.7.

Table 6.7: *Continuous Recognition Accuracy: Diphone System Bootstrapped from monophones with Additional MAP Estimation*

Experiment	Correct	Subs.	Deletions	Insertions	Accuracy
Diphones built from basic Monophones	54.10%	32.83%	13.07%	08.43%	46.10%
After additional MAP Estimation training	65.15%	29.17%	05.68%	15.48%	47.88%

The number of densities and the decoding times for each experiment are shown in Table 6.8.

Table 6.8: *Decoding Execution Time: Diphone System Bootstrapped from monophones with Additional MAP Estimation (Execution Time is depicted as hh:mm:ss.ms) with additional value in terms of real-time (RT)*

Experiment	Number of Densities	Execution Time
Diphones built from basic Monophones	3979	10:07:27.10 (Approx. 2.9 times slower than RT)
After additional MAP estimation training	3979	10:08:48.48 (Approx. 2.9 times slower than RT)

Interpretation

As expected, the diphones built from monophones yield the same phoneme sequences and therefore the same recognition accuracy when properly handled during decoding. This methodology is a very time-consuming way of duplicating the results of the efficient monophone-based system. The reason decoding takes in excess of 10 hours (60 times the time it takes the monophone-based system) is due to the fact that the computer ran out of system memory and was forced to make use of swap files. Constant switching of memory contents between the RAM modules and the harddrive has a detrimental effect on system

performance. The reason behind the depletion of memory resources is the combination of a large number of acoustic models and the large, well-trained mixture trees copied from the monophone models that are used as output probability density functions.

Despite the poor computational efficiency, it is shown that MAP estimation training improves system accuracy over the monophone-based recognition system.

6.3.4 Decision-tree Based State Clustering

Motivation

Decision-tree based state clustering is a popular method of optimally reducing the number of model parameters by means of state-level tying. Classification and Regression Trees (CART) are commonly used in conjunction with context-dependent models such as bi-phones and triphones to reduce the total parameter set size. In the case of diphones, states that model short segments of steady-state characteristics, such as the first and last states in the diphone model (equivalent to the central state of the monophone model), will be very similar across diphones that share that monophone segment. The output probability densities on these states can therefore be combined and shared with minimal loss in modelling capability. Reducing the parameter set size will not only improve computational efficiency, but accuracy as well.

Experimental Setup

The training data set is re-aligned using the three model sets trained in the experiments described in Sections 6.3.2 and 6.3.3: First approximation diphones after diphthong splitting (set A), diphones built from monophones (set B) and diphones built from monophones with additional MAP estimation training (set C). These labelled data sets are used to train CART trees – starting with the root node containing all possible classes and splitting subsequent nodes based on a question set. The leaf nodes are considered the new class set and a map file is used to map old diphone classes to the new clustered classes. This map file can then be used to relabel the data set and model specification files used to train the new set of models.

The CART clustering procedure involves setting a parameter called the State Occupancy (SO) parameter, which aims at controlling the eventual number of leaf nodes. The state occupancy parameter sets a threshold for the minimum number of datapoints represented by a leaf node. With a low SO parameter value, the CART tree can perform more splitting iterations, resulting in a larger tree. The experiments were performed at various values of the SO parameter for all three model sets.

As with all preceding experiments, the output probability densities are tree-based diagonal-covariance Gaussian mixtures. The mixture sizes and HMM topology depends on the model set, with values similar to the individual experiments preceding the CART experiments. All training and estimation procedures are exactly as with monophone models.

Results

Figure 6.4 shows the results of each of the three model sets for different values of the state occupancy parameter. Recognition accuracy is shown in parallel with execution time to indicate the common trend of improved recognition accuracy with a decrease in parameter set size. Figure 6.5 shows the same results as a function of the number of densities in each experiment for all three model sets.

The state occupancy value yielding the highest system accuracy was 3000 for set A, 2000 for set B and 1000 for set C. However, due to the low system requirements for the eventual diphone system, the best-attempt system for set C was chosen as the one with a state occupancy value of 2000. This system has slightly decreased accuracy, but is much more computationally efficient. The breakdown of evaluation results of these best-attempt systems for each model set is shown in Table 6.9.

Table 6.9: *Continuous Recognition Accuracy: Best results for each of the three diphone-based systems used in the CART experiment*

Experiment	Correct	Subs.	Deletions	Insertions	Accuracy
(A) First Approximation Diphones	69.86%	26.03%	04.11%	18.48%	48.11%
(B) Diphones built from Monophones	67.42%	26.43%	06.15%	15.09%	50.74%
(C) Diphones after additional MAP estimation	70.69%	24.69%	04.61%	15.51%	53.18%

The number of densities and decoding times for the best-attempt system for each model set is shown in Table 6.10.

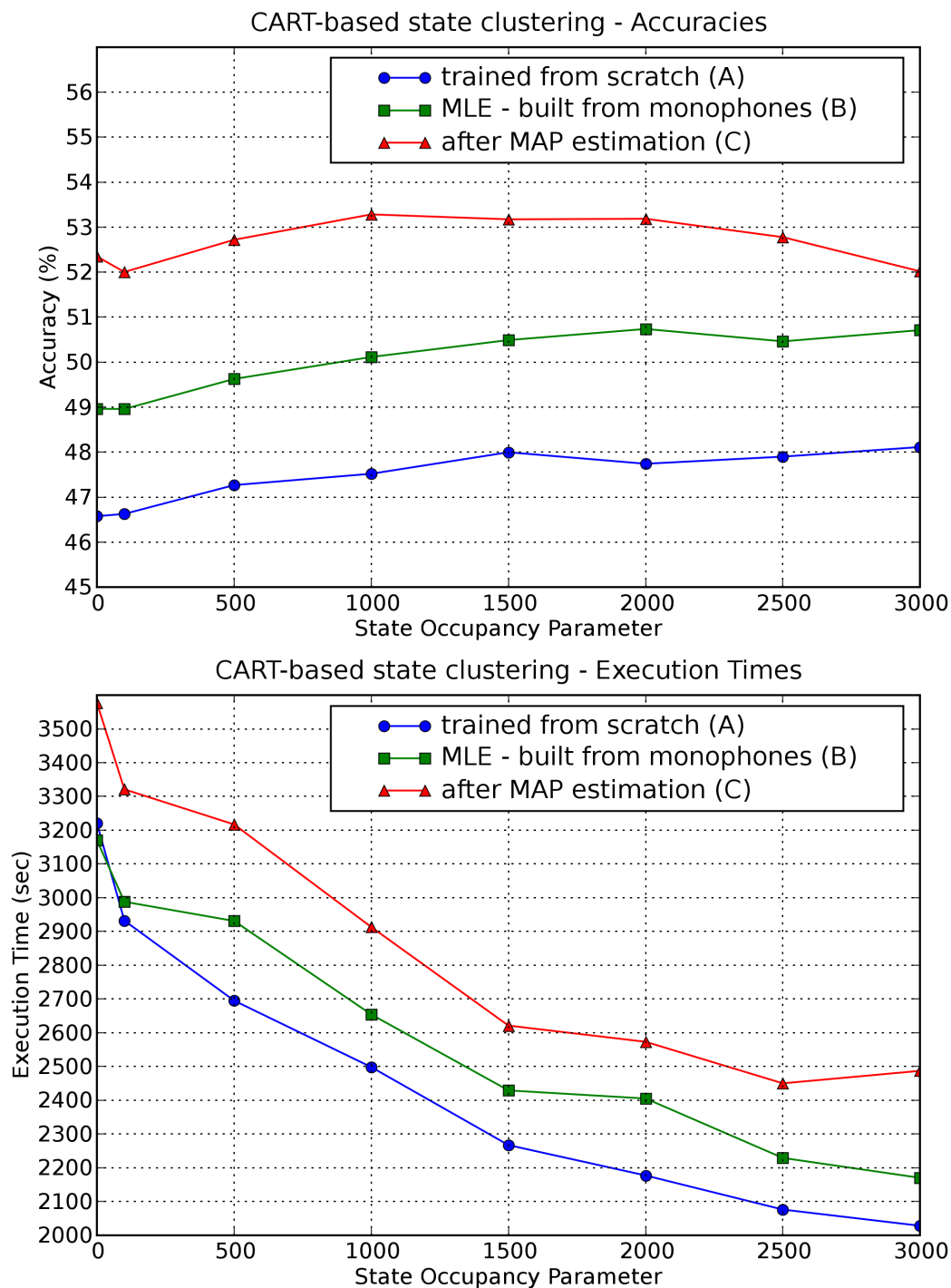


Figure 6.4: Accuracies and execution times of model sets A, B and C after decision tree-based state clustering at different state occupancy intervals. The optimal system is one yielding high recognition accuracy with the lowest possible number of parameters, resulting in faster decoding.

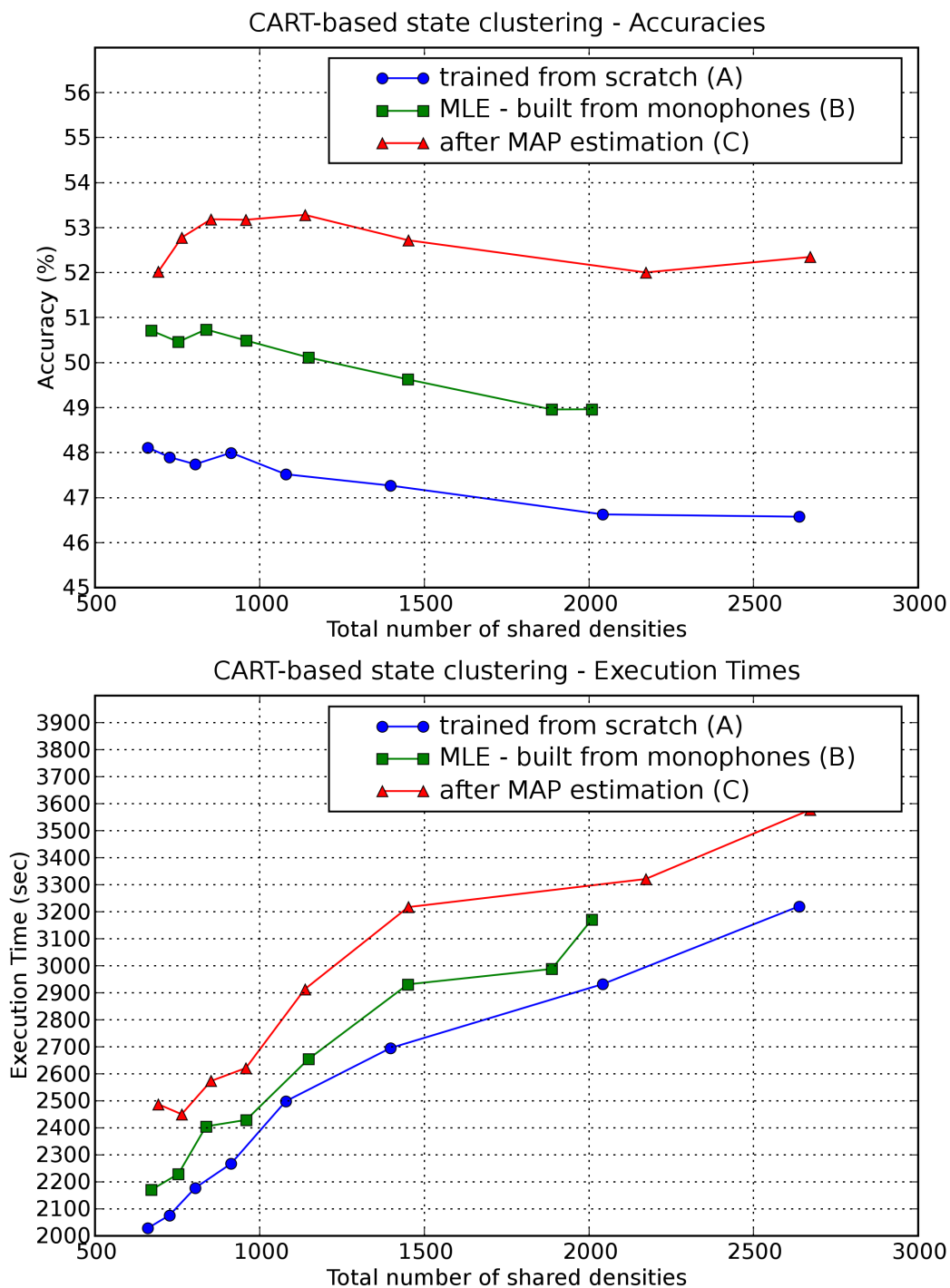


Figure 6.5: Accuracies and execution times of model sets A, B and C after decision tree-based state clustering as a function of the eventual density set size.

Table 6.10: *Decoding Execution Time of best results for each of the three diphone-based systems used in the CART experiment (Execution Time is depicted as hh:mm:ss.ms) with additional value in terms of real-time (RT)*

Experiment	Number of Densities	Execution Time
(A) First Approximation Diphones	660	00:33:48.18 (Approx. 6.4 times faster than RT)
(B) Diphones built from basic Monophones	837	00:40:04.61 (Approx. 5.3 times faster than RT)
(C) Diphones after additional MAP estimation	852	00:42:53.39 (Approx. 4.9 times faster than RT)

Interpretation

The CART experiments have brought about considerable improvement in system accuracy after decision-tree based state clustering. All three model sets have shown that the parameter set size can be significantly reduced before saturation occurs where subsequent clustering starts lowering recognition accuracy.

According to the results, the optimal diphone-based recognition system to use contains diphone models built from well-trained monophone models (reduced set), re-estimated via additional MAP estimation training and state-level tied to contain a total of about 850 densities. The optimal diphone-based recognition system operates at roughly 5 times faster than real-time, which although not as fast as a monophone-based system, is still very practical. Through the use of the matched-pairs test, the difference between the optimal diphone-based system and the monophone-based baseline system is statistically significant with the probability of the relative improvement being accidental extremely close to zero.

6.3.5 Interpretation of Diphone Results

The results of the diphone-based continuous phoneme recognition experiments show that diphones can achieve a higher recognition accuracy than the monophone-based baseline system with a reasonable reduction in computational efficiency. Accuracy is increased by 7%, the percentage of correctly recognised phonemes is increased by 16.5%

and the percentage of substitutions is decreased by 8%. These results also show that the diphone-based system can be improved further with decoding techniques aimed at limiting phoneme insertions, such as insertion penalties and biased loopback probabilities in the decoder. Ideally, the number of deletions and insertions should be balanced to provide decoded phoneme sequences that are roughly the same length as the correct sequences, but these requirements may vary with different applications according to the perceived severity of either an insertion or deletion error. The diphone-based system developed in this research would benefit greatly from explicit insertion error reduction.

The diphone-based recognition system is roughly 4 times slower than the monophone-based baseline system and has approximately 6 times as many parameters. Because the diphone-based system can still decode faster than real-time, it remains a practical solution for continuous-speech recognition. The CART experiments have shown that the total number of system parameters can be greatly reduced with minimal loss in recognition accuracy. In the experiments described above, the clustering procedure was used at different intervals of the state occupancy parameter to find the global maximum system accuracy. Initially recognition accuracy is increased when the number of parameters is decreased due to better class representation in the data set and better parameter estimation, but with subsequent clustering the recognition accuracy starts to decline because the parameter reduction is lowering the modelling capabilities of the diphone models. Therefore, the clustering procedure can be adjusted to find a diphone-based recognition system that is not as accurate as it can be, but has a significantly reduced parameter set that approaches the decoding efficiency of the simpler monophone-based system.

These experiments have shown that the adaptation techniques described in Chapter 5 are well suited for use in diphone-based phoneme recognition systems. Diphthong splitting provides a way to reduce the number of system parameters, increase computational efficiency and increase the system accuracy without any loss in modelling capability. By building the diphone models from well-estimated monophone models to use as a prior estimation in further MAP estimation training, well-estimated diphone models are obtained that is well suited to the input data at maximum modelling complexity. The effectiveness of this technique will depend on the accuracy of the monophone models and the amount of data used to fine-tune the diphone parameters. Decision-tree based state clustering is an effective way to increase system accuracy through parameter-tying and increased class representation in the data set. The largest gain in recognition accuracy is achieved when well-estimated diphone models are used as a base for the CART clustering procedure. By using these adaptation techniques to build on one another, a very good diphone-based system can be built that will outperform monophone-based systems with a reasonable

reduction in computational efficiency.

A general observation which is true for many practical applications, called the Pareto principle, states that *roughly 80% of the effects come from 20% of the causes*. This can also be applied to speech recognition results where a small percentage of the acoustic models usually causes a large percentage of the errors made during recognition. Therefore, focusing on improving this small subset of models may lead to significant overall improvement.

- **Substitutions:**

In the best-case diphone-based recognition system, the most problematic phoneme is the alveolar flap /d\/, which has a correctly recognised percentage of 11%. This phoneme both rarely occurs in the data set and is a short transitional sound that often sounds very similar to other sounds in continuous speech. The alveolar flap is most often confused with the alveolar trill (/r\/ as in *roost*), for example when the word *muddy* is pronounced quickly as *murry*. Being able to more accurately distinguish between these two sounds will require the incorporation of morphological knowledge as is typically used in word recognition. Duration modelling may also improve the recognition accuracy of the alveolar flap as it is typically slightly shorter than most other phonemes.

- **Insertions:** The largest percentage of insertions come from the two acoustic models that are not phonemes – SIL and OTHER. Both these models have special HMM topologies, and as such may be preferred by the decoder when choosing the optimal path through the spotter model. Insertions are typically handled with insertion penalties. Focussing on penalising the insertion of silence or garbage in the recognised phoneme sequence may drastically reduce the insertion rate.

- **Deletions:** The majority of deletions are caused by shortened vowels becoming syllabic consonants in continuous speech. For example, the neutral vowel /ə/ disappears in the word *button*, when it is pronounced /bʌtn/. This is a common problem when handling continuous speech and can also be handled by incorporating morphological knowledge into the decoding process.

6.4 Biphone-based Continuous Phoneme Recognition

6.4.1 Motivation

The result that diphone-based recognition systems achieve higher accuracy than monophone-based recognition systems is to be expected. The real test of diphone practicality is how

it compares with context-dependent models such as biphones and triphones. Biphones are subword units commonly used when incorporating contextual information, due to a relatively easy conversion from the simple and elegant monophone-based systems to biphone-based systems. Diphones can be used as an alternative methodology to context-dependent modelling techniques and are similar to biphones in many ways because both look at two phonemes at a time. The model set requirements are therefore similar and the two systems are directly comparable. For the purpose of this research both systems are analysed for use in low-resource environments to facilitate fast, accurate phoneme recognition.

For the best comparison, the experiments were designed to replicate the conditions used for the best possible diphone system described in Section 6.3. Well-trained monophone models are used in the second system to facilitate additional MAP estimation training, after which decision-tree based state clustering is used to reduce the parameter set size. The biphone-based system is ultimately reduced to approximate the parameter-set size used in the best diphone system.

6.4.2 Experimental Setup

The experiments done in this section pertain to left-context biphones. In biphone-based systems it is rarely necessary to use both left- and right-context biphones simultaneously as this constitutes information duplication and doubles the final number of acoustic models. Previous biphone-based phoneme recognition experiments done on the AST data set (English subset) have shown very similar results for left- and right-context biphones [29]. As this thesis is not directly focused on the evaluation of biphone-based systems to obtain the best possible configuration, it was deemed appropriate to only provide the results for one of the two contexts. Left-context biphone models assume that the most important influence on the current phoneme is the one directly preceding it, whereas right-context biphone models assume that the most important influence on the current phoneme is the one following it.

The original phoneme-aligned data set is relabelled to contain left-context biphone labels. The monophone boundaries remain stationary while the biphone labels are created according to the phonemic context. For left-context biphone models the first phoneme in an utterance is labelled as being preceded by the phoneme SIL (silence) to elegantly handle the end-point of the biphone sequence.

Because of the inherent differences between diphones and biphones it is not possible to duplicate all adaptation techniques exactly. However, whenever an adjustment was made to a technique employed on the biphone system, the aim was to improve the biphone

system as much as possible, so as to not invalidate the eventual comparison between the systems.

It is extremely important to use the basic monophone models for the diphone-based systems to enable direct comparison of each intermediate step with the monophone baseline system. However, to make the comparison between the diphone-based system and the biphone-based system more useful, the biphones were built from an optimised monophone set with higher recognition accuracy (48.6%) than the monophones used for the diphone-based system. The optimised monophone models utilise additional training iterations with interim realignment of the original training set and use a lexicon to do simultaneous alignment of multiple segmentation levels for better model re-estimation. The same monophone optimisation techniques could not be easily applied to the diphone-based system. Additionally, for the purpose of this research, these monophone model improvements do not reveal any important differences between monophone- and diphone-based systems and therefore do not add value to the objective analysis of the diphone-based system.

Using the improved monophone model set, the segmentation boundaries of the biphone models (which are at the same locations as the corresponding phoneme boundaries) are more robust and better positioned in the data set. The diphone boundaries are already robust due to the fact that they are situated in the steady-state portions of the phonemes. It was thought that, if the diphone-based system could out-perform an improved, but still very similar biphone-based system, this would be a much better result.

The characteristics inherent in a diphone-based system makes it perfect for the diphthong splitting adaptation technique because it removes redundancy from a system already used to model transitions. Because biphone models do not explicitly model phoneme transitions diphthongs must still be included in the model set. This means that the biphone-based system can have at most 44^2 (1936) acoustic models, whereas the diphone-based system can have at most 34^2 (1156) acoustic models.

The output probability densities are tree-based diagonal-covariance Gaussian mixtures with a maximum size of 256 for all biphone states, including the HMM states representing sections of the monophones SIL and OTHER.

All training and estimation procedures are exactly as with monophone and diphone models. A variation of the diphone spotter is used as a biphone decoder because of similar constraints on biphone model ordering.

The experiments done on biphone-based systems were designed to be able to closely compare biphone results with diphone results. As far as possible the biphone system was treated exactly the same as the equivalent best diphone system. The experiments can be divided into the following phases:

1. **Building from monophones and preclustering**

A good starting point for biphone models is to clone their state output probability densities from well-estimated monophone models, which can then be used as prior parameters for additional MAP estimation training. Unfortunately the combination of large well-estimated state probability densities and the large number of acoustic models quickly depleted the memory resources. Even with the use of swap files and 4GB system memory, the physical limits of our current hardware were reached. The reason this problem was not encountered with diphone models (that has a similar model set size), is due to the use of diphthong splitting to reduce the monophone set size used as a base. This technique could not be applied to biphone models. To circumvent this problem a preclustering phase was introduced in which CART trees were used to reduce the parameter set size just enough in order to fit into system memory.

Preclustering is done by using the monophone-aligned biphone-labelled data set to train a CART tree. Ideally the state-level clustering should be done in such a way that the new clustered models can still be built from monophone models. The clustering should therefore take place on state level, between models of the same phoneme in different contexts, and not between states belonging to different phonemes. What had to be taken into account was that current limitations in the software implementation of CART trees in the PatrecII software suite do not enable manual branching decisions to be made. Node splitting within the tree is done by means of a question set from which the CART algorithm chooses the question that best discriminates between subsequent classes. Separate trees can be trained for each set of biphones by modelling the same phoneme in different contexts, but then there would be no global measure of when the ideal parameter set size has been achieved. As a workaround to this problem, a normal unconstrained CART tree was trained and then “unfolded” for classes grouping states together that come from two different monophones. The state occupancy parameter is adjusted to “overcluster” in order to eventually obtain the correct number of classes. The new classes represent the new set of densities cloned from its equivalent in a monophone model and used in the construction of a new set of HMM models – one for each biphone. These biphone HMM models are evaluated by creating a biphone decoder similar to those used in diphone-based systems.

2. **MAP estimation training**

The models that were built from a preclustered set of well-trained monophone densities are used as prior densities in additional MAP estimation training. The MAP

training is done on a relabelled data set reflecting the new classes obtained through preclustering.

3. Decision-tree based State Clustering

The last step in creating the biphone-based system is the final state-level clustering used to reduce the total number of system parameters to a level comparable with the best possible diphone-based system. The CART algorithm is used exactly as described for diphone-based systems.

6.4.3 Results

The breakdown of evaluation results for each left-context biphone-based recognition system is summarised in Table 6.11.

Table 6.11: *Continuous Recognition Accuracy: Recognition systems based on left-context biphone models*

Experiment	Correct	Subs.	Deletions	Insertions	Accuracy
Built from improved monophone models after preclustering	67.36%	27.86%	04.78%	16.44%	48.62%
After additional MAP estimation training	67.53%	27.68%	04.79%	16.36%	48.90%
Decision-tree based state clustering	67.54%	26.40%	06.06%	14.63%	51.44%

The best-case left-context biphone-based system has 1149 densities and decodes the testing data set in approximately 49 minutes, which is approximately 4.3 times faster than real-time (RT).

6.4.4 Interpretation

According to the results, the left-context biphone-based system that was built from well-estimated monophone models, re-estimated via additional MAP estimation training and clustered with decision-tree based state clustering techniques achieves a higher recognition accuracy than the equivalent monophone-based system. The biphone-based recognition system operates at roughly 4 times faster than real-time, which although not as fast as a monophone-based system, is still very practical. Through the use of the matched-pairs

test, the difference between the biphone-based system and both the monophone-based baseline system and best-case diphone-based system is statistically significant with the probability of the relative improvement being accidental extremely close to zero.

6.5 Comparison of Systems in Limited-Resource Environments

The results of the biphone-based phoneme recognition experiment shows that the diphone-based system is the better choice when evaluated under similar conditions in a low-resource environment.

Initially the monophone-built biphones perform better than the corresponding monophone-built diphones as well as the monophone-baseline system due to the fact that the biphones were built from slightly optimised monophone models. These results only reflect the differences between the monophones used as base for each of the two systems, because no additional training is done after building the diphone and biphone models from monophone models. The decoder also ensures that each system produces the exact same phoneme sequence as the monophone-based system it was based upon.

After MAP estimation both the diphone- and biphone-based systems achieve slightly higher recognition accuracies than the systems directly built from monophones. The diphone-based system accuracy increased with approximately 1.8%, whereas the biphone-based system accuracy increased with approximately 0.3%. This is because the diphone-based system can better utilise the additional MAP estimation training due to the fact that there are more free parameters that can be fine-tuned to the transitional characteristics of the diphone. In a biphone model, the parameters on one side of the phoneme are very specific to the context, whereas the parameters on the other side of the phoneme are very generic and common to all contexts of that specific phoneme. These generic parameters will not change during MAP estimation training as they already closely approximate the monophone parameters. For biphones only the parameters modelling the context of the phoneme will be fine-tuned during the MAP estimation training. The MAP estimation procedure is therefore more beneficial for diphone-based systems than for biphone-based systems.

The final results after state clustering show that, although the biphone-based system was built from optimised monophone models, the diphone-based system outperforms it in similar parametric conditions. The best-case biphone-based system has 1149 densities and achieve a recognition accuracy of 51.44%. The best-case diphone-based system has 852 densities and achieve a recognition accuracy of 53.18%. In the CART experiments done

in Section 6.3.5, the diphone-based system with similar parametric conditions than the biphone-based system had 1137 densities and achieved a recognition accuracy of 53.28%. Therefore, when the same adaptation techniques used for the diphones are applied to the biphone-based system, even when the biphone-based system started with a better set of monophone models, the diphone-based system outperforms the biphone-based system when evaluated in similar parametric conditions.

6.6 Summary

The series of experiments described in this chapter was done in order to objectively analyse the use of diphones as subword units for speech recognition and focused on intermediate adaptation techniques that could improve the acoustic modelling of diphone units. The diphone results were compared to a monophone baseline system and a biphone-based system designed to utilise the same adaptation techniques as used for the diphones.

The chapter starts with a description of the experimental setup common to all experiments, including the hardware platform, software suite and the speech corpus used. The signal processing techniques used for front-end speech processing and the statistical modelling parameters used during hidden Markov model based acoustic modelling are also described. The system evaluation for phoneme recognition is discussed, as well as the matched-pairs tests designed to determine whether improved results are statistically significant.

The first experiments were done to obtain a monophone-based recognition system that could serve as a baseline for subsequent comparisons. The monophone-based system achieved a phoneme recognition accuracy of 46.1% and decoded at an impressive 22 times faster than real-time.

The first-approximation diphone experiment was designed to facilitate the conversion from the basic monophone-based system to one utilising diphone models, to enable us to identify potential problems early on. The issues associated with an exponentially larger parameter set caused the first approximation system to achieve only 13.2% accuracy, which emphasised the need to handle the data scarcity problem. The use of a diphone-structured spotter during decoding increased recognition accuracy to 37.8%. The first significant results were obtained after using diphthong splitting to decrease the total number of monophones the diphones are based on, which increased the recognition accuracy to 44.5%. Although this system is slightly less accurate than the monophone baseline system and much more inefficient, it served as a diphone-based framework for subsequent improvements in acoustic modelling efficacy.

Building diphone models from monophones and using the well-estimated parameters as priors for additional MAP estimation training increased phoneme recognition accuracy to 47.9%, the first improvement on the monophone-based baseline system. The biggest problem with this system is the long decoding time (approximately 3 times slower than real-time) due to the fact that the memory requirements are very high, so that additional swap space is needed. This issue is addressed with decision-tree based state clustering (CART), which also increases recognition accuracy due to better class representation in the data set. Three sets of CART experiments were done, in order to find the optimal value of the state occupancy parameter for the first-approximation system, the diphone-based system built from monophones, and the system built from monophones with additional MAP estimation training. The best result was achieved by the last system utilising all adaptation techniques discussed in Chapter 5. This best-case diphone-based system achieved a recognition accuracy of 53.2%, while decoding at approximately 5 times faster than real-time. While not as fast as the monophone-based system, the diphone-based system is still feasible in practical situations, with a good increase in recognition accuracy when compared to the monophone-based baseline system.

The last part of the chapter is dedicated to the comparison of the best-case diphone-based system with an analogous left-context biphone-based system utilising the same adaptation techniques as used for the diphone-based system. Even when biasing the biphone-based system by building the biphones from an improved monophone set, the diphone-based system outperforms the best-case biphone-based system (51.4%).

Chapter 7

Conclusions

7.1 Concluding Perspective

This thesis accomplished its goal of objectively analysing the use of diphones in hidden Markov Model-based continuous-speech recognition in a limited-resource environment. Diphone characteristics were examined in relation to the more commonly used context-dependent biphones and it was found that diphone-based systems face similar difficulties with regards to problems such as data scarcity, reliable parameter estimation, high resource requirements and the handling of unseen and inter-word contexts. Diphones were found to have more robust segmentation boundaries than context-dependent biphones due to the location of diphone boundaries within the short stationary part in the centre of phonemes. Diphones are also designed to tightly fit unique transitional characteristics so that there are relatively few parameters that are general to more than one diphone, whereas biphones have parameters that are generic to multiple contexts on the side of the phoneme that is not modelled in a specific left- or right-context. These observations were shown to be true in the experiments done in Chapter 6, where additional MAP estimation training with well-estimated monophone prior densities were more beneficial to the diphone-based system than the biphone-based system.

The main problems faced by diphone models are caused by the large parameter set size, which leads to low class representation in the data set, poorly estimated parameters and low computational efficiency. These problems were addressed through the use of adaptation techniques such as diphthong splitting, MAP estimation training and decision-tree based state clustering with the use of CART trees. Several experiments were also done to investigate the effect of different state occupancy parameters. By reducing the total number of parameters, higher class representation lead to better estimated models, increasing the system accuracy as well as reducing the system requirements. The

question was whether these techniques are more beneficial to diphone-based systems than other large-parameter systems, such as context-dependent biphone-based systems. The diphone-based system that gave the best results was compared to a biphone-based system utilising the same adaptation techniques. Both systems were executed on the same computer platform. Even with a slight bias in favour of the biphone models (being based on better estimated monophone models) the diphone-based system outperforms the equivalent biphone-based system with a margin of approximately 2%.

The main advantages of using diphones as subword units is the inherent co-articulation modelling and the fact that a limited number of parameters can be used more efficiently to accurately model unique characteristics in phoneme transitions. The main disadvantages of using diphones as subword units are the high system requirements and data scarcity that accompany large parameter sets, as well as high model complexity. The procedure required to adapt a simple recognition system based on phonemes to one based on the transitions between phonemes is also relatively intricate as adjustments to both lexical and language modelling are necessary to facilitate word recognition.

7.2 Context Within Existing Research

The results of the experiments done for this thesis are analogous to results achieved in three cases of diphone research done in the 1990's.

The conclusion of research conducted in Poland by Basztura *et al.* [4] in 1998 into the use of diphones as subword units for hidden Markov model-based automatic continuous-speech recognition, was that a diphone-based system using a hybrid HMM/ANN algorithm lead to 9% higher word recognition accuracy in relation to comparable experiments that used phonemes as basic units.

Fissore *et al.* [32] showed in 1996 that a hidden Markov model system based on a set of stationary and transitional units achieved an average increase in word recognition accuracy of 2.5% over a system using both biphone- and triphone models in similar parametric conditions. Further, Dobrišek *et al.* [27] directly compared transitional acoustic models with context-dependent phone models by analysing HMM-based recognition systems based on either diphones, biphones or triphones with the same number of model parameters to enable a direct comparison. Their results showed that diphones achieved an average increase in word recognition accuracy of 2.2% when compared to biphones.

In our research the phoneme recognition accuracy achieved by the best diphone-based system is 7.2% higher than that for similar monophone-based systems and approximately 2% higher than the accuracy found for a comparable left-context biphone-based system.

Calculations of word recognition accuracy are outside the scope of this thesis but the relative increase in recognition accuracy achieved by the diphone-based system is still analogous to the previous research done.

7.3 Future Work

This research served as a basic proof of concept that diphones are good subword units to use for phoneme recognition in low resource environments within the framework of a hidden Markov model based continuous-speech recognition system.

7.3.1 Diphone-based System

There are a number of further refinements that can be made to the diphone-based system that are not addressed by this thesis. The following improvements are suggested:

- For a large part of this research diphone models were treated as special cases of phoneme models. The diphone-based recognition system would greatly benefit from in-depth analyses of the spectral and temporal characteristics inherent in phoneme transitions to be able to make informed decisions as to the best HMM topology and state output probability distributions to use for each diphone model. As there are so many diphone models, this would require either a significant effort from a research point of view or the development of an observation system that can be used to automatically find the optimal HMM settings for each diphone model. These settings would include the number of states, the transitions between these states, the size of the tree-based mixtures used as output probability distributions on each HMM state and the type of distribution used in each case, as dictated by the availability of suitable training data.
- Results indicate that the spotter configuration used during decoding has a significant impact on the system accuracy. The 0-gram grammar used in the current diphone-based system can be replaced with a bi-gram grammar model based on counting the number of occurrences of each diphone. For phonemes, weighing the probability of transitioning to a certain phoneme according to its frequency in the training set results in a mono-gram grammar, with a bi-gram grammar achieved by counting the number of transitions between two specific phonemes. Therefore, implementing a mono-gram grammar on diphone-level equates to a bi-gram grammar on phoneme-level. The grammatical model used during decoding is highly sensitive

to the training data set. To represent a complete picture of relative diphone occurrences in a given language it may have to be trained on an external textual data set if the available training data is inadequate.

- With an adapted diphone-based lexicon and the handling of inter-word transitions, language modelling can be used to evaluate word recognition performance. Continuous-speech recognition systems typically show large improvements in recognition performance when complex language models are used, but adapting this system to directly utilise diphones as subword units instead of first translating recognised diphone sequences to phoneme sequences will require further research. Because modelling a language as a sequence of probable transitions is more natural than modelling it as a sequence of sounds, a language model based on diphones may have an advantage over similar language models based on phonemes. Most practical speech recognition systems use language models for word recognition and evaluating diphones on this level will ultimately prove the practicality of the diphone-based system, or lack thereof.

7.3.2 Comparison with a Triphone-based System

The comparison of a diphone-based system with a biphone-based system is directly applicable due to the fact that both diphones and biphones aim to model the interaction between two phonemes. Triphones provide a method for increasing the context-modelling capabilities by modelling the interaction between three phonemes, which is equivalent to two adjacent diphones. Therefore, comparing a diphone-based system to a triphone-based system can only provide an overview of their relative performance given the current conditions. A direct comparison between a diphone-based system and one based on triphones would be more applicable if the diphone-based system modelled two adjacent diphones.

Theoretically, triphone-based systems will always outperform the simpler systems given that

- enough training data with sufficient representation for each triphone is available for reliable parameter estimation,
- system requirements, such as memory usage, is unrestricted and
- there are no time-constraints as to how long the system is allowed to decode.

Previous research suggests that a diphone-based recognition system can outperform a triphone-based system if they are evaluated in similar parametric conditions [27], but is yet to be proved within the current experimental setup.

The following experiments can be used to investigate the performance of the diphone-based system relative to a triphone-based system:

- Triphone-based recognition systems commonly start by using decision-tree based state clustering to reduce the total number of system parameters and ensure sufficient class representation. In low-resource environments the result is a significant reduction in system parameters and consequent modelling capabilities, which may make the triphone-based system less effective than biphone- or diphone-based systems. Relabelling the monophone aligned data set to triphone labels and using decision-tree based state clustering to reduce the total number of parameters to the same number used in the diphone and biphone experiments can facilitate a rough comparison of these systems.
- Ideally the diphone-based system should be compared to a triphone-based system that has received the exact same treatment and can therefore be directly compared. As with the biphone-based system, the triphone models can be cloned from monophones and used as prior models for additional MAP estimation training before using decision-tree based state clustering to reduce the number of parameters. Because of the vast number of possible triphone models in a given language, this training will have to be done on a system containing excessive amounts of system memory and computing power, preferably a distributed computing system.

Bibliography

- [1] “Nationmaster encyclopedia - artificial neural network.” Internet Article.
<http://www.nationmaster.com/encyclopedia/Artificial-neural-network>.
- [2] BAHL, L., BROWN, P., DE SOUZA, P., and MERCER, R., “Maximum mutual information estimation of hidden Markov model parameters for speech recognition.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 1986*, 1986.
- [3] BAHL, L. R., JELINEK, F., and MERCER, R. L., “A maximum likelihood approach to continuous speech recognition.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1983, Vol. 5, pp. 179–190.
- [4] BASZTURA, C., LISIAK, P., and STARONIEWICZ, P., “Automatic Speech Recognition based on Diphones.” in *9th Mediterranean Electrotechnical Conference: MELECON '98*, vol. 1, (Tel-Aviv, Israel), pp. 6–10, May 1998.
- [5] BAUM, L. E. and EAGON, J. A., “An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology.” *Bulletin of the American Mathematical Society*, 1966, Vol. 73, No. 3, pp. 360 – 363.
- [6] BAUM, L. E. and PETRIE, T., “Statistical Inference for Probabilistic Functions of Finite State Markov Chains.” *The Annals of Mathematical Statistics*, 1966, Vol. 37, pp. 1554–1563.
- [7] BAUM, L. E., PETRIE, T., SOULES, G., and WEISS, N., “A Maximisation Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains.” *The Annals of Mathematical Statistics*, February 1970, Vol. 41, No. 1, pp. 164–171.
- [8] BEULEN, K. and NEY, H., “Automatic Question Generation for Decision Tree Based State Tying.” in *Acoustics, Speech and Signal Processing - Proceedings of*

- IEEE International Conference: ICASSP 1998*, (Seattle, Washington, USA), May 1998.
- [9] BISHOP, C. M., *Pattern Recognition and Machine Learning*. 233 Spring Street, New York, NY 10013, USA: Springer Science+Business Media, LLC, 2006.
- [10] BLOMBERG, M., “Creating Unseen Triphones by Phone Concatenation in the Spectral, Cepstral and Formant Domains.” in *Conference Proceedings of the European Conference on Speech Communication and Technology: EUROSPEECH*, 1997.
- [11] BLOOTHOOFT, G., HAZAN, V., HUBER, D., and LLISTERRI, J., “European studies in phonetics and speech communication.” Internet: <http://www.haskins.yale.edu/Reprints/HL0962.pdf>, 1995. Haskins Laboratories, Connecticut.
- [12] BRAND, R., “A Comparison of Gaussian Mixture Variants with Application to Automatic Phoneme Recognition.” Master’s thesis, University of Stellenbosch, 2007.
- [13] BREIMAN, L., FRIEDMAN, J., OLSHEN, R., and STONE, C., *Classification and Regression Trees*. New York: Chapman & Hall, 1993.
- [14] BRUGNARA, F., “Model Agglomeration for Context-dependent Acoustic Modeling.” in *Conference Proceedings of the European Conference on Speech Communication and Technology: EUROSPEECH*, (Aalborg, Denmark), September 2001.
- [15] CHEN, S. S. and GOPINATH, R. A., “Model Selection in Acoustic Modelling.” in *Conference Proceedings of the 6th European Conference on Speech Communication and Technology: EUROSPEECH*, vol. 3, (Budapest, Hungary), pp. 1087–1090, September 1999.
- [16] CHESTA, C., LAFACE, P., and RAVERA, F., “Bottom-up and Top-down State Clustering for Robust Acoustic Modeling.” in *Conference Proceedings of the European Conference on Speech Communication and Technology: EUROSPEECH*, 1997.
- [17] CHOMSKY, N., *Syntactic Structures*. The Hague/Paris: Mouton, 1957.
- [18] CILLIERS, F. D., “Tree-based Gaussian Mixture Models for Speaker Verification.” Master’s thesis, University of Stellenbosch, 2005.

- [19] COLLA, A. and ROSENBERG, A., “Unsupervised Bootstrapping of Diphone-like Templates for Connected Speech Recognition.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 1987*, vol. 12, (Dallas, Texas, USA), pp. 1281 – 1284, April 1987.
- [20] COLLA, A. and SCIARRA, D., “Automatic Diphone Bootstrapping for Speaker-adaptive Continuous Speech Recognition.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 1984*, vol. 3, (San Diego, California, USA), pp. 35.2.1 – 35.2.4, March 1984.
- [21] COMMUNICATIONS, N., “Dragon naturallyspeaking 10.” Internet: <http://www.nuance.com/naturallyspeaking/resources/default.asp>. [23 Feb 2009].
- [22] DE IPIÑA, K. L., VARONA, A., TORRES, I., and RODRIGUEZ, L. J., “Decision Trees for Inter-word Context Dependencies in Spanish Continuous Speech Recognition Tasks.” in *Conference Proceedings of the European Conference on Speech Communication and Technology: EUROSPEECH*, vol. 2, (Budapest, Hungary), pp. 899 – 902, September 1999.
- [23] DE’ATH, G. and FABRICIUS, K. E., “Classification and Regression Trees: A Powerful yet Simple Technique for Ecological Data Analysis.” *Ecology*, November 2000, Vol. 81, pp. 3178–3192.
- [24] DELLER, J. R., PROAKIS, J. G., and HANSEN, J. G., *Discrete Time Processing of Speech Signals*. Upper Saddle River, New Jersey, USA: Prentice Hall PTR, 1993.
- [25] DEMPSTER, A. P., LAIRD, N. M., and RUBIN, D. B., “Maximum likelihood from incomplete data via the EM algorithm.” *Journal of the Royal Statistical Society. Series B (Methodological)*, 1977, Vol. 39, No. 1, pp. 1–38.
- [26] DEROO, O., RIS, C., and DUPONT, S., “Context Dependent Hybrid HMM/ANN Systems for Large Vocabulary Continuous Speech Recognition System.” in *Conference Proceedings of the 6th European Conference on Speech Communication and Technology: EUROSPEECH*, vol. 3, (Budapest, Hungary), pp. 1095–1098, September 1999.
- [27] DOBRISEK, S., MIHELIC, F., and PAVESIC, N., “Acoustical Modelling of Phone Transitions: Biphones and Diphones – What are the Differences?.” in *Proceedings of 6th European Conference on Speech Communication and Technology*:

- EUROSPEECH*, (Budapest, Hungary), pp. 1307–1310, ISCA Archive, September 1999.
- [28] DUPONT, S., RIS, C., DEROO, O., FONTAINE, V., BOITE, J., and ZANONI, L., “Context Independent and Context Dependent Hybrid HMM/ANN Systems for Vocabulary Independent Tasks.” in *Conference Proceedings of the European Conference on Speech Communication and Technology: EUROSPEECH*, (Rhodes, Greece), pp. 1947–1950, September 1997.
- [29] ENGELBRECHT, H. A., “Automatic Phoneme Recognition of South African English.” Master’s thesis, University of Stellenbosch, 2004.
- [30] FANT, G., “Phonetics and Phonology in the last 50 years.” in *From Sound to Sense: International Conference held at MIT (2004)*, June 2004.
- [31] FANT, G. and MARTONY, J., “Instrumentation for Parametric Synthesis (OVE II).” *Speech Transmission Laboratory - Quarterly Progress and Status Report, KTH*, 1962, Vol. 3, No. 2, pp. 18–24. <http://www.speech.kth.se/qpsr>.
- [32] FISSORE, L., LAFACE, P., MICCA, G., and RAVERA, F., “Vocabulary Independent Acoustic-Phonetic Modelling for Continuous Speech Recognition.” in *Proceedings of European Signal Processing Conference: Eusipco '96*, pp. 1615 – 1618, 1996.
- [33] FRITSCH, J., “Mixture Trees – Hierarchically Tied Mixture Densities for Modelling HMM Emission Probabilities.” in *Conference Proceedings of the 6th European Conference on Speech Communication and Technology: EUROSPEECH*, vol. 3, (Budapest, Hungary), pp. 1103–1106, September 1999.
- [34] GAO, S. and LEE, C.-H., “A Discriminative Decision Tree Learning Approach to Acoustic Modeling.” in *Conference Proceedings of the European Conference on Speech Communication and Technology: EUROSPEECH*, (Geneva), pp. 1833 – 1836, 2003.
- [35] GAUVAIN, J. L. and LEE, C.-H., “Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains.” *IEEE Transactions on Speech and Audio Processing*, 1994, Vol. 2, pp. 291–298.
- [36] GHITZA, O. and SONDHI, M., “Hidden Markov Models with Templates as Non-stationary States: An Application to Speech Recognition.” *Computer Speech and Language*, April 1993, Vol. 7, No. 2, pp. 101 – 119.

- [37] GILLICK, L. and COX, S. J., “Some Statistical Issues in the Comparison of Speech Recognition Algorithms.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 1989*, (Glasgow), pp. 532–535, 1989.
- [38] GLASS, J. and HAZEN, T., “Telephone-based Conversational Speech Recognition in the Jupiter Domain.” in *International Conference on Speech and Language Processing: ICSLP 1998*, (Sydney, Australia), December 1998.
- [39] GREENWOOD, A. R., “Articulatory Speech Synthesis using Diphone Units.” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing: ICASSP 1997*, vol. 3, (Munich), pp. 1635–1638, April 1997.
- [40] HASTIE, T. and STUETZLE, W., “Principal Curves.” *Journal of the American Statistical Association*, 1989, Vol. 84, No. 406, pp. 502–516.
- [41] HATON, J.-P. (Ed.), *Fundamentals in Computer Understanding: speech and vision*. Trumpington Street, Cambridge: Cambridge University Press, 1987.
- [42] HOWELL, P., “The Extent of coarticulatory effects: Implications for models of speech recognition.” *Speech Communication*, July 1983, Vol. 2, No. 2–3, pp. 159–163.
- [43] HWANG, M.-Y. and HUANG, X., “Hidden Markov models for speech recognition.” 1991. Internet:
`ftp://reports.adm.cs.cmu.edu/usr/anon/1991/CMU-CS-91-124.ps`.
- [44] JELINEK, F., “Continuous speech recognition by statistical methods.” *Proceedings of the IEEE*, 1976, Vol. 64, No. 4, pp. 532–556.
- [45] JITSUHIRO, T., MATSUI, T., and NAKAMURA, S., “Automatic Generation of Non-uniform HMM Topologies Based on the MDL Criterion.” *IEICE Transactions on Information Systems (The Institute of Electronics, Information and Communication Engineers)*, 2004, Vol. E87-D, No. 8, pp. 2121–2129.
- [46] LEE, C.-H., RABINER, L. R., PIERACCINI, R., and WILPON, J. G., “Acoustic Modeling of Subword Units for Speech Recognition.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 1990*, vol. 4, pp. 721–724, April 1990.
- [47] LEE, G. and LEE, J., “Integrated Speech and Morphological Processing in a Connectionist Continuous Speech Understanding for Korean.” *Computing Research Repository (CoRR)*, 1996. Internet: <http://arxiv.org/abs/cmp-lg/9603005>.

- [48] LEVINSON, S. E., *Fundamentals in Computer Understanding: speech and vision*, Ch. Statistical Methods for Speaker Independence. Cambridge University Press, 1987.
- [49] LEVINSON, S. E., RABINER, L. R., and SONDHI, M. M., “An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition.” *Bell System Technical Journal*, 1983, Vol. 62, No. 4, pp. 1035–1074.
- [50] LEWIS, R. J., “An Introduction to Classification and Regression Tree (CART) Analysis.” *Annual Meeting of the Society for Academic Emergency Medicine*, 2000.
- [51] LOUW, P. H., ROUX, J. C., and BOTHA, E. C., “African Speech Technology (AST) Telephone Speech Databases: Corpus Design and Contents.” in *Conference Proceedings of the European Conference on Speech Communication and Technology: EUROSPEECH*, vol. 3, (Aalborg, Denmark), pp. 2055 – 2058, September 2001.
- [52] MALFRERE, F. and DUTOIT, T., “High-quality Speech Synthesis for Phonetic Speech Segmentation.” in *Conference Proceedings of the European Conference on Speech Communication and Technology: EUROSPEECH*, 1997.
- [53] MARINO, J. and NOGUEIRAS, A., “Top-down Bottom-up Hybrid Clustering Algorithm For Acoustic-phonetic Modeling of Speech.” in *Conference Proceedings of the European Conference on Speech Communication and Technology: EUROSPEECH*, vol. 3, (Budapest, Hungary), pp. 1343 – 1346, September 1999.
- [54] MARINO, J., PACHES-LEAL, P., and NOGUEIRAS, A., “The Demiphone Versus the Triphone in a Decision-tree State-tying Framework.” in *International Conference on Speech and Language Processing: ICSLP 1998*, (Sydney, Australia), December 1998.
- [55] MITKOV, R. (Ed.), *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2003.
- [56] MURTHY, S. K., “Automatic construction of decision trees from data: A multi-disciplinary survey.” *Data Mining and Knowledge Discovery*, 1998, Vol. 2, pp. 345–389.
- [57] NADAS, A., MERCER, R. L., BAHL, L. R., BAKIS, R., COHEN, P. S., COLE, A. G., JELINEK, F., and LEWIS, B. L., “Continuous Speech Recognition with Automatically Selected Acoustic Prototypes Obtained by Either Bootstrapping

- or Clustering.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 1981*, (Atlanta, Georgia), pp. 1153–1155, 1981.
- [58] NAKAMURA, M., IWANO, K., and FURUI, S., “Differences between Acoustic Characteristics of Spontaneous and Read Speech and their Effects on Speech Recognition Performance.” *Computer Speech and Language*, 2008, Vol. 22, pp. 171 – 184.
- [59] NIESLER, T. and LOUW, P., “Comparative Phonetic Analysis and Phoneme Recognition for Afrikaans, English and Xhosa using the African Speech Technology Telephone Speech Databases.” *South African Computer Journal*, June 2004, No. 32, pp. 3 – 12.
- [60] O’NEILL, P., VASEGHI, S., DOHERTY, B., TAN, W. H., and MCCOURT, P., “Multi-phone Strings as Subword Units for Speech Recognition.” in *International Conference on Speech and Language Processing: ICSLP 1998*, (Sydney, Australia), December 1998.
- [61] OTTENHEIMER, H. J., *The Anthropology Of Language: An Introduction To Linguistic Anthropology*. Wadsworth, 2005.
- [62] PADMANABHAN, M., JAN, E. E., BAHL, L. R., and PICHENY, M., “Decision-tree based Feature-space Quantization for Fast Gaussian Computation.” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 325–330, 1997.
- [63] PALLETT, D. S., FISHER, W. M., and FISCUS, J. G., “Tools for the Analysis of Benchmark Speech Recognition Tests.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 1990*, pp. 97–100, 1990.
- [64] RABINER, L. R., “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.” *Proceedings of the IEEE*, 1989, Vol. 77, No. 2, pp. 257–286.
- [65] RABINER, L. R. and JUANG, B. H., “An Introduction to Hidden Markov Models.” *IEEE ASSP Magazine*, 1986, pp. 4–17.
- [66] RABINER, L. R. and JUANG, B. H., *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey: Prentice Hall, 1993.
- [67] RABINER, L. R., JUANG, B. H., LEVINSON, S. E., and SONDHI, M. M., “Recognition of isolated digits using hidden Markov models with continuous

- mixture densities.” *Bell Systems Technical Journal*, 1985, Vol. 64, No. 6, pp. 1211–1234.
- [68] RABINER, L. R. and SCHAFER, R. W., *Digital Processing of Speech Signals*. Englewood Cliffs, New Jersey: Prentice Hall, 1987.
- [69] REINHARD, K. and NIRANJAN, M., “Non-linear Speech Transition Visualization.” in *IEEE International Conference on Artificial Neural Networks*, (Cambridge, UK), July 1997.
- [70] REINHARD, K. and NIRANJAN, M., “Parametric Subspace Modelling of Speech Transitions.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 1998*, vol. 2, (Seattle, Washington, USA), pp. 1105 – 1108, May 1998.
- [71] REINHARD, K. and NIRANJAN, M., “Diphone Multi-trajectory Subspace Models.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 1999*, pp. 1001 – 1004, March 1999.
- [72] REINHARD, K. and NIRANJAN, M., “Diphone Subspace Models for Phone-based HMM Complementation.” in *Conference Proceedings of the 6th European Conference on Speech Communication and Technology: EUROSPEECH*, vol. 3, (Budapest, Hungary), pp. 1351–1354, September 1999.
- [73] REINHARD, K. and NIRANJAN, M., “Matched Filter Design for Diphone Subspace Models.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 2000*, (Istanbul, Turkey), pp. 3430 – 3433, 2000.
- [74] RICHARDSON, M., BILMES, J., and DIORIO, C., “Hidden-articulator Markov Models for Speech Recognition.” *Speech Communication*, 2003, Vol. 41, pp. 511–529.
- [75] RILEY, M. D. and LJOLJE, A., “Automatic Generation of Detailed Pronunciation Lexicons.” in *Automatic Speech and Speaker Recognition* (LEE, C. and SOONG, F. K. (Eds)), 1996.
- [76] ROACH, P., “A little encyclopaedia of phonetics.” Internet: <http://www.personal.rdg.ac.uk/~llsroach/encyc.pdf>, 2002. [5 Dec 2008].
- [77] RUMELHART, D. E., HINTON, G. E., and WILLIAMS, R. J., “Learning Internal Representations by Error Propagation.” 1986, pp. 318–362.

- [78] SCAGLIOLA, C., “Continuous Speech Recognition Without Segmentation: Two Ways of Using Diphones as Basic Speech Units.” *Speech Communication*, July 1983, Vol. 2, No. 2-3, pp. 199–201.
- [79] SCAGLIOLA, C. and MARMI, L., “Continuous Speech Recognition via Diphone Spotting – A Preliminary Implementation.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 1982*, vol. 7, pp. 2008 – 2011, May 1982.
- [80] SHANNON, C. E., “A Mathematical Theory of Communication.” *The Bell System Technical Journal*, 1948, Vol. 27, pp. 379–423 and 623–656.
- [81] TYAGI, V. and WELLEKENS, C., “On desensitizing the Mel-Cepstrum to spurious spectral components for Robust Speech Recognition.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 2005*, vol. 1, pp. 529–532, 2005.
- [82] VAIR, C., MERCOGLIANO, M., and FISSORE, L., “Incremental Training of CDHMMs using Bayesian Learning.” in *Conference Proceedings of the European Conference on Speech Communication and Technology: EUROSPEECH*, vol. 6, (Budapest, Hungary), pp. 2753 – 2756, September 1999.
- [83] VAN SANTEN, J. P. H. and SPROAT, R. W., “High-accuracy Automatic Segmentation.” in *Conference Proceedings of the 6th European Conference on Speech Communication and Technology: EUROSPEECH*, vol. 6, (Budapest, Hungary), pp. 2809–2812, September 1999.
- [84] WAX, M., “Construction of Tree Structured Classifiers by the MDL Principle.” in *Acoustics, Speech and Signal Processing - Proceedings of IEEE International Conference: ICASSP 1990*, vol. 4, pp. 2157–2160, April 1990.
- [85] XU, W., DUCHATEAU, J., DEMUYNCK, K., DOLOGLOU, I., WAMBACQ, P., COMPERNOLLE, D. V., and HAMME, H. V., “Accuracy versus Complexity in Context-dependent Phone Modelling.” in *Conference Proceedings of the 6th European Conference on Speech Communication and Technology: EUROSPEECH*, vol. 3, (Budapest, Hungary), pp. 1127–1130, September 1999.

Appendix A

Selected Topics from Linguistic Theory

A.1 International Phonetic Alphabet

Below Table A.1 contains a list of phonemes used in South African English in the standard IPA notation, the equivalent notation used for monophones in this research and example words to explain the usage of each phoneme. This is a reduced set, representing only sounds that can distinguish between words in the English language, by grouping allophones (similar sounding phones) together. For example the phoneme /æ/ represents both the short (*act*) and longer (*bad*) versions of the same sound. These different phones do not distinguish meaning in the English language, but differ due to their morphological context. Phones with longer duration are often noted by adding a text-length mark ([æ:]). Complete information on the full IPA for English can be found on the International Phonetic Association website (<http://www.arts.gla.ac.uk/IPA/>).

Table A.1: Phoneme Chart : Vowel and Consonant Sounds for South African English

IPA Phoneme notation	Monophone used in research	Example
<i>Vowels and Diphthongs</i>		
/æ/	ae	act, bad
/ə/	@	ago, potter
/ɜ:/	@	burn, fern
/ʌ/	^	up, luck
/ɒ/	Q	hot, moth

Table A.1 - continued from previous page

IPA Phoneme notation	Monophone used in research	Example
/ɑ:/	Q	car, heart
/ɔ:/	O	saw, door
/ɛ/	E	pet, lair
/ɪ/, /i:/	i	cosy, eat
/ʊ/, /u:/	u	put, good
/əɪ/	@i	day, tape
/əʊ/	@u	go, note
/aɪ/	^i	sky, alive
/aʊ/	^u	cow, out
/ɔɪ/	Oi	boy, choice
/iə/	i@	beer, near
/ɪʊ/	iu	new, municipal
/ʊə/	u@	tour, poor
<i>Consonants</i>		
/p/	p	pack, slap
/b/	b	bow, lab
/t/	t	time, bit
/d/	d	den, bed
/ɾ/	d\	muddy (alveolar flap)
/k/	k	cow, look
/g/	g	give, log
/f/	f	fan, laugh
/v/	v	van, voice
/w/	w	water, awake
/s/	s	send, mess
/m/	m	man, room
/n/	n	nice, land
/ŋ/	N	sing, blank
/l/	l	leg, bell
/ɹ/, /r/	r	ring, battery
/r/	r\	roost (alveolar trill)
/h/	h	hat, behind
/z/	z	zebra, berserk

Table A.1 - continued from previous page

IPA Phoneme notation	Monophone used in research	Example
/j/	j	yet, beyond
/θ/	T	thin , math
/ð/	D	that , brother
/ʃ/	S	shop , blush
/z/	Z	genre, leisure
/tʃ/	tS	chew , chop
/dʒ/	dZ	jump , jaw
<i>Non-speech</i>		
silence	SIL	
noise/coughing/laughing	OTHER	

A.2 Types of Phonemes

These and other linguistic definitions can be found in an online publication by Professor P Roach [76].

Short and Long Vowels

Vowels are the set of sounds made when the vocal tract offer the least amount of resistance to the flow of air and sound. The vowel sounds differ due to differing positions of the front, middle and back of the tongue, as well as the shape of the lips (compare the rounded sound of the /u/ vowel in *good* with the open sound of the /I/ vowel in *greed*). Vowels are usually found in the centre of a syllable with distinctions made between the shorter and longer versions of the same sound determined by its context. In speech recognition short and long vowels of the same sound are often grouped together when duration modelling can be used to distinguish between them, reducing the total number of phoneme models necessary.

Diphthongs

Diphthongs are phonemes that are characterised by a glide from one vowel to another. They can either be classified as combinations of phonemes or as phonemes in their own

right. The latter is useful when the two sounds effectively merge during continuous speech.

Plosives

Plosives are a type of consonant created by completely obstructing the flow of air through the vocal tract and releasing the built-up air with a sharp sound followed by aspiration. Examples include voiced (/b/ in *bin*) and unvoiced (/t/ in *tin*) plosives with articulation occurring in different parts of the vocal tract.

Fricatives

Fricatives are a type of consonant created by forcing air through a very narrow gap in the vocal tract. Examples include voiced (/z/ in *zap*; /v/ in *van*) and unvoiced (/s/ in *sap*; /f/ in *fan*) fricatives.

Nasals

Nasals are consonants created by obstructing airflow through the oral cavity and redirecting it through the nose. They are always voiced consonants, for example /m/ in *map*, /n/ in *nap* and /ŋ/ in *bang*.

Approximants

Approximants are consonants that are created by a limited amount of airflow obstruction, ranging from consonants that are close to vowels (/w/ in *wet*) to liquids (/l/ in *lip*) that only partially constrict airflow without producing fricative noise.

Affricates

Affricates are consonants consisting of a plosive followed by a fricative with the same place of articulation. The two affricates used in the English language are /tʃ/ in *child* and /dʒ/ in *jolly*.

A.3 Additional Terms Related to the Production of Speech Sounds

Aspiration

The sound produced when vocal restrictions in the vocal tract are released and air is allowed to escape relatively freely. Aspiration typically occurs after plosives, fricatives and affricates in certain conditions. For example, the plosive /p/ is aspirated in the word *pack*, but not in the word *spoon*.

Rounding phonemes

Rounding refers to the shape of the lips and mouth cavity to produce a rounded sound. Usually rounded sounds are vowels, but certain consonants such /w/ in *wet* are also rounded to different degrees.

Fortis and lenis consonants

Fortis (“strong”) and lenis (“weak”) are words used to describe consonants in terms of the amount of energy used to produce each sound. In English the distinction between fortis and lenis consonants are often taken as voiced and unvoiced.

Coronal sounds

Coronal sounds are sounds that are produced by an articulation configuration where the blade of the tongue is raised from its normal resting position, such as /t/ in *top*.

Anterior consonant

Anterior consonants are the group of consonants articulated in the front part of the mouth, such as /b/ in *bus*.

Syllabic consonant

In continuous speech, the vowels in some syllables can become weak or lost. For example the word *button* is usually pronounced /bʌtn/ in continuous speech instead of /bʌtən/, leaving the second syllable with no discernable vowel. These weak vowels are also called syllabic consonants, due to the heavy influence of the neighbouring consonants on the sound articulation.

Stridents

Stridents are strong and clearly audible fricatives, containing high amounts of energy. For example the fricative /S/ in *shin* is a positive strident, whereas the fricative /T/ in *thin* is a negative strident.

Appendix B

Speech Corpus

B.1 The African Speech Technology (AST) Speech Corpus

The African Speech Technology project was a 3-year project aiming to promote the development of the official languages of South Africa through language and speech technology applications [51, 59]. The project was a joint venture between linguistic specialists, speech recognition and generation experts and consultants from various universities in South Africa, which was successfully completed in March 2004. The ultimate goal was the development of a toolkit for multilingual speech recognition applications, such as telephone booking systems. Full technical reports are available on the website (<http://www.ast.sun.ac.za>).

B.1.1 Data sets

The AST speech corpus consists of databases for five of the eleven official South African languages – English, Afrikaans, Xhosa, Zulu and southern Sotho. The English database contains 5 subsets, each representing a different speech variety as spoken by mother-tongue and non-mother-tongue speakers. By incorporating non-native speech varieties, the database provides better representation of the multilingual character of the South African populace.

B.1.2 Collection Parameters

The speech data was collected from both fixed and mobile telephone networks (roughly equal amounts) and consists of a mixture of read and spontaneous speech. Each individual speaker provided 38 to 40 utterances, resulting in call durations of between 7

and 10 minutes each. The utterances contain read items such as isolated digits, as well as digit strings, money amounts, dates, times, spellings and phonetically rich words and sentences. Spontaneous items include gender, age, home language, place of residence and level of education.

The speech signals are 8-bit mono, a-law encoded speech waveforms obtained at a sampling rate of 8kHz. The databases also include phonemically and orthographically aligned transcriptions of each utterance.

B.1.3 Phoneme Set

The original phoneme set used to transcribe the AST data contained 154 phonemes, based on the definitions of the International Phonetic Alphabet. For the purpose of this research the phoneme set was reduced to a minimal set of 44 phonemes, by grouping allophones and acoustically similar phonemes together. The reduced phoneme set is described in appendix A.

B.2 Subword Unit Statistics

B.2.1 Monophones

The training set of the English database contains a total of 8597 utterances from 1054 unique speakers. The testing set contains a total of 901 utterances from 112 unique speakers that do not overlap with the speakers from the training set. The total number of occurrences of each phoneme and their average durations are shown in table B.1.

Table B.1: Occurrence and Duration statistics for monophone labels found in the training and testing subsets of the English AST Data set

Phon	Training set		Testing set	
	Occ.	Mean Dur(s)	Occ.	Mean Dur(s)
SIL	74380	0.2605	7983	0.258706
n	56531	0.1274	5816	0.127043
i	52532	0.1352	5443	0.136951
@	50447	0.1094	5333	0.109409
t	45901	0.1210	4873	0.121276
s	39942	0.1301	4188	0.130869

Table B.1 - continued from previous page

Phon	Training set		Testing set	
	Occ.	Mean Dur(s)	Occ.	Mean Dur(s)
E	22167	0.1405	2372	0.140551
f	20896	0.1364	2183	0.136388
l	18294	0.1203	1928	0.120787
k	17850	0.1205	1848	0.120006
d	17526	0.1172	1908	0.119428
@i	16469	0.1556	1725	0.155282
^i	16004	0.1460	1700	0.145926
r	15040	0.1256	1679	0.125805
u	14814	0.1303	1568	0.130041
w	14207	0.1276	1495	0.126921
Q	14174	0.1273	1472	0.126714
^	13977	0.1300	1510	0.130112
v	13612	0.1284	1459	0.126812
@u	13544	0.1726	1390	0.171673
m	12700	0.1171	1294	0.114616
ae	10446	0.1169	1077	0.115749
T	10210	0.1263	1046	0.127876
O	9364	0.1347	952	0.134039
OTHER	9034	0.1622	970	0.162562
b	8709	0.1235	939	0.123217
p	8514	0.1136	878	0.11542
z	8064	0.1207	785	0.119305
j	7569	0.1401	778	0.144177
r\	7094	0.1339	656	0.136316
D	5702	0.1037	630	0.101901
h	4815	0.1069	474	0.107429
N	4002	0.1025	388	0.103445
g	3646	0.1071	357	0.106358
^u	3316	0.1212	361	0.120606
S	3130	0.1121	302	0.11264
i@	3031	0.1480	281	0.148021
dZ	2720	0.1371	263	0.138102

Table B.1 - continued from previous page

Phon	Training set		Testing set	
	Occ.	Mean Dur(s)	Occ.	Mean Dur(s)
tS	2668	0.1397	275	0.134923
iu	928	0.1215	110	0.125595
d\	703	0.1297	56	0.120882
Oi	356	0.1200	29	0.126006
u@	239	0.1109	38	0.106607
Z	142	0.1098	12	0.119967

Appendix C

CART

C.1 Question Set

The question set is usually manually constructed according to a set of phonetic classes, the type of input data and subword unit used in the speech recognition system, although there are algorithms for automatic question set generation [8]. The optimal question to use for a specific node is determined by the splitting criterion leading to the highest gain in overall classification accuracy. The class labels in the subword model set have a specific structure consisting of a number of fields with delimiting characters between them. The CART questions are based on possible classifications of these fields.

Diphones

Diphone labels are defined as “s&s_s” where the first two fields denote the two phonemes in the transition and the third field describes the HMM state number. For example, the first state of the HMM model for diphone “m&æ” is the class “m&æ_0”.

Biphones

Left-context biphone labels are defined as “s-s_s” where the first field denotes the phoneme preceding the current phoneme, the second field denotes the current phoneme and the third field describes the HMM state number. For example, the first state of the HMM model for the left-context biphone “m-æ” is the class “m-æ_0”.

Right-context biphone labels are defined as “s+s_s” where the first field denotes the current phoneme, the second field denotes the phoneme following the current phoneme and the third field describes the HMM state number. For example, the first state of the HMM model for the right-context biphone “æ+t” is the class “æ+t_0”.

Complete Question set for Three Fields

The question set used during decision-tree based state clustering in this research starts by defining a number of set definitions. These sets represent logical groupings of similar sounds according to their phonemic characteristics and are listed in table C.1.

For each field in the subword structure and each set definition, a question is created to determine whether that particular field of the current class is contained within the set (for example “FIELD 1 IN Rounded_Vowel”). Questions are also created to determine whether a particular field is a certain phoneme (for example “FIELD 1 IS @”).

Questions regarding the state field only pertain to the current state number (for example “FIELD 3 IS 0”).

Table C.1: Set definitions used in CART question set

<i>Vowels and Diphthongs</i>	
High Vowel	i, @, u
Medium Vowel	ae, ^, @, E, @i, @u
Low Vowel	Q, ae, ^, O, ^u, ^i, Oi
Rounded Vowel	O, @u, Oi, u, w
Unrounded Vowel	Q, ae, ^, ^u, @, ^i, E, @, @i, h, i, l, r, j
Reduced Vowel	@
I Vowel	@, i
E Vowel	E, @i
A Vowel	Q, ae, ^u, ^i, @
O Vowel	O, @u, Oi
U Vowel	^, @, u
<i>Consonants</i>	
Unvoiced Consonant	tS, f, h, k, p, s, S, t, T
Voiced Consonant	b, d, D, g, dZ, l, m, n, N, r, v, w, j
Front Consonant	b, f, m, p, v, w
Central Consonant	d, D, d\, l, n, r, s, t, T, z, Z
Back Consonant	tS, g, h, dZ, k, N, S, j
Fortis Consonant	tS, f, k, p, s, S, t, T
Lenis Consonant	b, d, D, g, dZ, v, z, Z
Non-Fortis/Lenis	h, l, m, n, N, r, w, j
Coronal Consonant	tS, d, D, dZ, l, n, r, s, S, t, T, z, Z
Non Coronal	b, f, g, h, k, m, N, p, v, w, j

Anterior Consonant	b, d, D, f, l, m, n, p, s, t, T, v, w, z
Non Anterior	tS, g, h, dZ, k, N, r, S, j, Z
Continuent	D, f, h, l, m, n, N, r, s, S, T, v, w, j, z, Z
No Continuent	b, tS, d, g, dZ, k, p, t
Positive Strident	tS, dZ, s, S, z, Z
Negative Strident	D, f, h, T, v,
Neutral Strident	b, d, g, k, l, m, n, N, p, r, t, w, j
Syllabic Consonant	@
Voiced Stop	b, d, g
Unvoiced Stop	p, t, k
Front Stop	b, p
Central Stop	d, t
Back Stop	g, k
Voiced Fricative	tS, D, v, z, Z
Unvoiced Fricative	tS, f, s, S, T
Front Fricative	f, v
Central Fricative	D, s, T, z
Back Fricative	tS, dZ, S, Z
Affricate Consonant	tS, dZ
Non-Affricate	D, f, s, S, T, v, z, Z
<i>General</i>	
Silence	SIL

C.2 Pruning

The greedy top-down induction method of building CART trees is used to grow the tree from a root node containing all training data and iteratively splitting the data on each node of the tree until one or more of the following termination conditions has been reached:

- The state occupancy falls below a preset threshold. The state occupancy is a measure of the number of datapoints left on a given tree node. A node where all possible splits will result in children nodes with too few datapoints will be considered a leaf node and not split any further.

- The measure of overall improvement used during the splitting process is below a certain minimum for all possible questions at a specific node. It is therefore not beneficial to the system to split the node any further according to any of the available questions.

The splitting threshold value is often set very low to encourage aggressive tree growth at first. Later the tree is pruned back to an optimal size, thus balancing the model complexity and generality of the CART tree. This strategy is necessary to counter the phenomenon that some nodes may experience a time of little gain in overall system performance only to produce large gains after several more splits. The greedy induction method performs node splits with no regard to subsequent splits, creating a tree that is not globally optimal, but piecewise locally optimal.

Pruning is also necessary to avoid over-fitting the tree on the random idiosyncrasies or noise that may be present in the training data. Generality is important to ensure that the model is robust enough to effectively handle unseen data. The tree should not provide such a perfect fit to the training data that it might not at all be beneficial to unseen data. Pruning is often done by evaluating the trained CART tree with an accurately labeled evaluation data set, that was not used in the training process. During this phase, splits in the original tree that were too closely modelled on individual training data is pruned back if those splits are not general enough to properly fit the evaluation data.

An alternative strategy is to use top-down bottom-up hybrid clustering algorithms to get the best properties of both top-down and bottom-up tree growing approaches [16, 53].

C.3 Minimum Description Length Based Induction and Pruning

An effective tree growing strategy often used is based on the *Minimum Description Length* (MDL) principle. The MDL principle is rooted in information theory, stating that the best splitting model is the one that yields the shortest description length of the data set. Because we are summarising the training data by evaluating the most important descriptive input parameters that yield the best class separation, the splitting model that is able to lead to the shortest summary is the best corresponding model to use. In this scenario, every possible split is regarded as a competitive model for the training data configuration obtained directly after the split, and the model with the minimal description length for the class labels on the child nodes is considered the

optimal splitting criterion. The mathematical description of MDL for use in decision trees is taken from [84].

Given an internal node with N datapoints, each labelled as belonging to one of a set of J classes, let

$$Y = \{y_1, y_2, \dots, y_N\} \quad y_n \in J \quad (\text{C.1})$$

denote the set of class labels attached to the N datapoints on the node, and N_j the number of datapoints belonging to class j . The description length of the node is defined as

$$\text{MDL}\{Y\} = N h(p_1) + \frac{1}{2} \log \frac{N}{2(p_1 p_2 \dots p_J)} \quad (\text{C.2})$$

where

$$p_j = \frac{N_j}{N} \quad (\text{C.3})$$

denotes the relative frequency of class j and $h(p)$ is the entropy function

$$h(p) = -p \log(p) - (1 - p) \log(1 - p) \quad (\text{C.4})$$

However, to evaluate splitting criteria, the description length is not calculated for the node itself, but for the resulting split denoted by a specific question. Therefore let $Y_L^{(i)}$ represent the class labels of the datapoints on the left child after applying splitting criterion i , and $Y_R^{(i)}$ represent the class labels of the datapoints on the right child. Let Q represent the set of all possible questions to be considered as splitting criteria. The optimal splitting criterion \hat{i} is identified as the one leading to the minimum description length of the two child nodes, or

$$\hat{i} = \arg \min_i \text{MDL}\{Y_L^{(i)}, Y_R^{(i)}\} \quad \forall i \in Q \quad (\text{C.5})$$

The question of whether to continue splitting a node into child nodes or not can also be addressed by using the MDL principle, effectively negating the need for pruning. If the description length of the proposed child nodes are greater than the description of the current node for all possible splitting criteria, splitting the node will result in a worse model than is currently available and should be avoided. Therefore, the current node is considered a leaf node if, and only if

$$\text{MDL}\{Y_L^{(i)}, Y_R^{(i)}\} > \text{MDL}\{Y\} \quad \forall i \in Q \quad (\text{C.6})$$