Speech Analyser in an ICAI System for TESOL

by

Huayang Xie

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Master of Science
in Computer Science.

Victoria University of Wellington
2004

Abstract

There is an increasing demand for computer software which can provide useful personalised feedback to English as a Second Language (ESL) speakers on prosodic aspects of their speech, to supplement the shortage of ESL teachers and reduce the cost of learning.

This thesis concentrates on constructing such an Intelligent Computer Aided Instruction (ICAI) prototype system, particularly focusing on one component — the Speech Analyser. The speech analyser recognises a user's speech, identifies the rhythmic stress pattern in the speech, discovers stress and rhythm errors in the speech, and provides reports for the other component generating personalised feedback to the user on ways of effectively improving the prosodic aspects of the speech.

We build an Hidden Markov Model (HMM) based speech recogniser to recognise a user's speech. A set of parameters for constructing the recogniser is investigated by an exhaustive experiment implemented in a client/server computing network. The exploration suggests that the choice of parameters is very important. We build stress detectors to detect the rhythmic stress pattern in the user's speech by using both Support Vector Machine (SVM) and Decision Tree (DT) techniques. The detector using SVM outperforms the one using DT. It suggests that SVM is more suitable for a relatively large data set with all numeric data than DT. We build an error identifier to automatically identify stress and rhythm errors in the user's speech. A two-layer phoneme alignment algorithm using the Needleman/Wunsch technique is developed to facilitate the prosodic error identification problem. Our study also suggests that the foot comparison method is better than Vowel Onset Point comparison method for automatically identifying the main rhythm errors in the user's speech.

Acknowledgments

I would like to give special thanks to my two great supervisors, Peter Andreae and Mengjie Zhang, for their time, patience, enlightening instructions, eye for detail, and everything they did for me, especially the financial support arrangement. During the time of doing my research, not only did they gave their time to discuss the progress of my thesis and were always available to solve problems, but also encouraged me and helped me to convert part of my work into two published papers ([82, 83]). Without their generous assistance, my research may have been a painful experience. I will never forget the great experience working with them.

Thanks also to other members of this research group, including David Crabbe and Paul Warren of the School of Linguistics and Applied Language Study, Irina Elgort of the University Teaching Development Center, Neil Leslie of School of Mathematical and Computing Sciences, and Mike Doig of VicLink.

Most importantly, I thank my parents for their constant encouragement. My wife Yi, although very busy with her own work, always gave my research a higher priority and provided substantive support. Especially during the last phase toward my thesis completion, my son Tianyi and my coming baby Tianying have given me great emotional support.

I also thank all the people not mentioned here who have contributed to my research for their help, including other AI group and Technical group members in School of Mathematical and Computing Sciences, and people from the virtual communities I have never met.

Thanks also to Simon Doherty for helpful advice and comments.

I was mainly financially supported by the New Economy Research Fund, New Zealand. I was also financially supported by the School of Mathematical and Computing Sciences Graduate Scholarship and Computer Science Studentship, Victoria University of Wellington, New Zealand.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis is about constructing an Intelligent Computer Aided Instruction (ICAI) system for Teaching English to Speakers of Other Languages (TESOL).

The ICAI system first analyses an English as a Second Language (ESL) learner's speech in order to identify errors in the speech. It then provides useful personalised feedback to the learner based on the errors.

This thesis mainly studies how to recognise an ESL learner's speech, how to detect the stress pattern in the speech, and how to automatically identify the stress and rhythm errors in the speech.

1.1 Motivation

Today, English is the most widely used language in the world. Many people from non-native English speaking countries learn English in order to effectively communicate with the world. Speaking is one important aspect of communication skills. Good speech involves many issues, such as vocabulary, grammar, and pronunciation. Among these issues, pronunciation, especially prosodic aspects in spontaneous speech, always remains the most serious problem.

Learning spoken English well requires lots of practice. However it is really hard for ESL speakers to compare their speech with native speakers' speech and identify prosodic problems by themselves given the large differences in sound features during practice. A great deal of personalised feedback to identify and correct prosodic errors in ESL speakers' speech must be provided in order to make the learning procedures more effective and productive. Providing personalised feedback from human teachers is very expensive and the shortage of ESL teachers is increasing. As more and more ESL students come to New Zealand, there is an increasing demand for computer software that can provide useful personalised feedback to ESL speakers on prosodic aspects of their speech to supplement the shortage of ESL teachers and reduce the cost of learning.

1.2 ICAI Outline

An outline of the ICAI system, which consists of two sub-systems — a Pedagogic Component (Peco) and a Speech Analyser (Span), is shown in Figure 1.1.

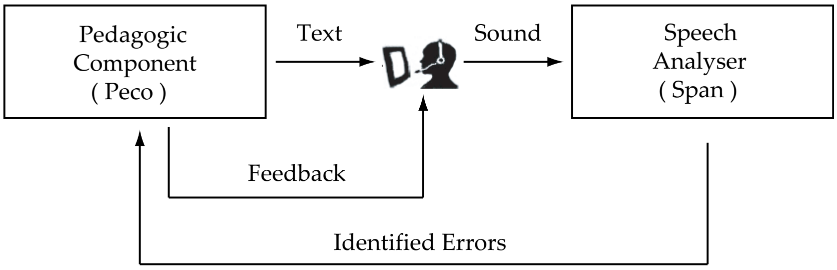


Figure 1.1: Outline of the ICAI system.

First of all, Peco displays a sentence on the screen for a user to read aloud. The recorded user's sound is then fed into Span. Figure 1.2 briefly outlines the elements and procedures involved with and in Span. More details will be covered in later chapters. Inside Span, a speech recognition process is applied to the sound and annotates/labels the sound at a

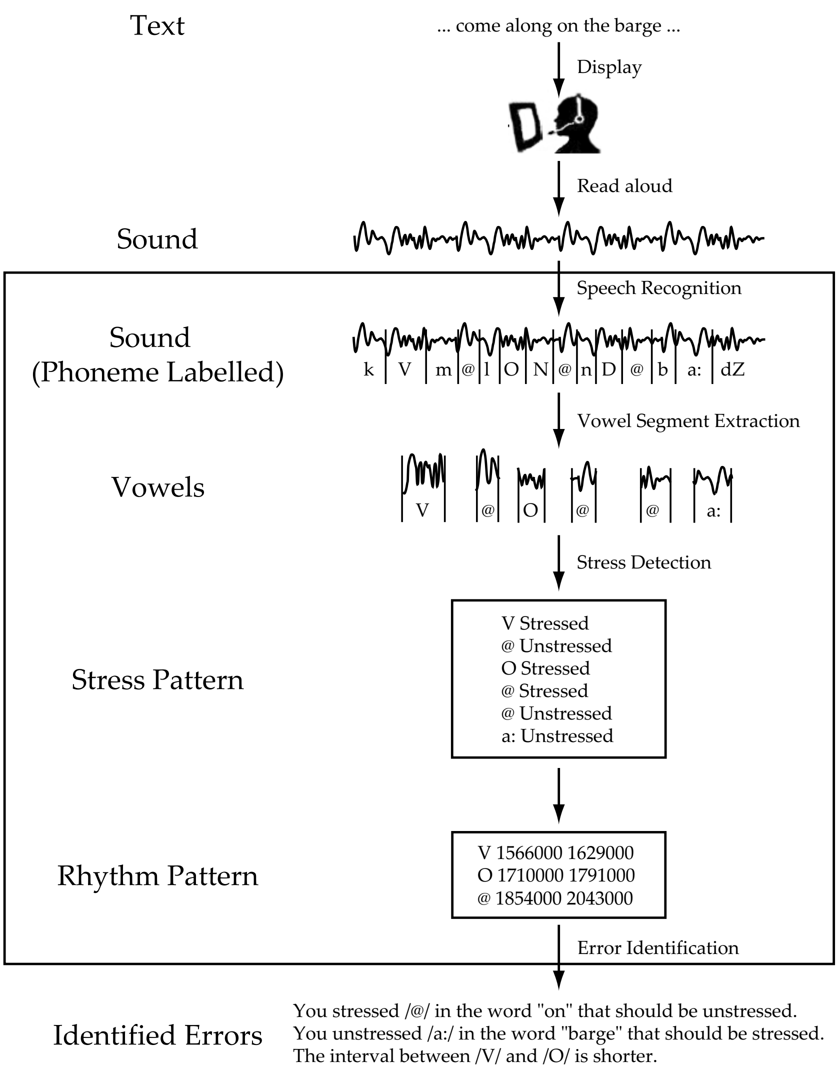


Figure 1.2: Elements and procedures with and in Span.

phoneme level. A sequence of vowel segments is then extracted from the phoneme labelled sound and is fed into a stress detection process to generate a stress pattern and a rhythm pattern. An error identification process is applied to the stress pattern and the rhythm pattern in order to ascertain stress and rhythm errors. Finally the identified errors are fed back to Peco, which provides personalised feedback to the user.

1.3 Span Overview

This section provides a detailed overview of Span as illustrated in Figure 1.3.

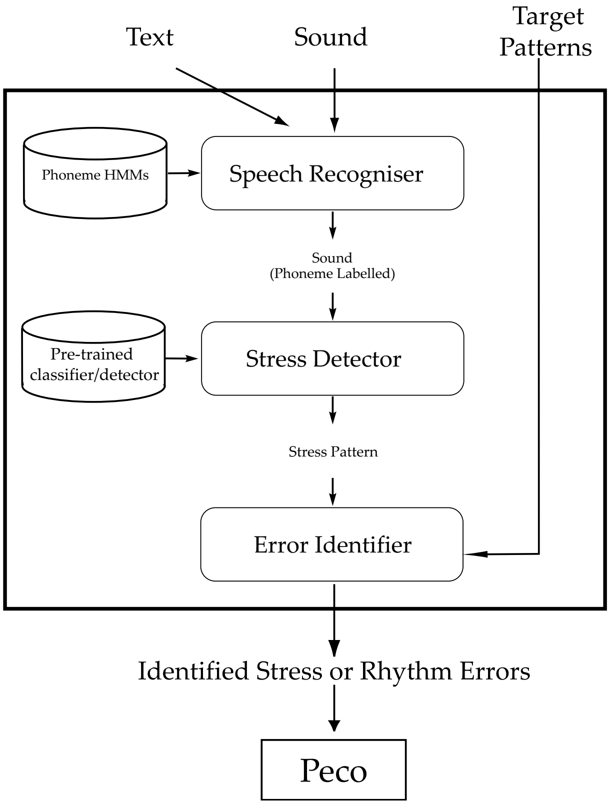


Figure 1.3: Overview of Span.

There are three inputs to Span, text, sound and target patterns. The text is a sequence of words of a sentence that a user reads aloud. The sound is uttered by the user and is encoded into the standard Pulse Code Modulation (PCM) format [12]. These two inputs are used immediately in Span. The target patterns include a stress pattern and a rhythm pattern in a native English speaker's speech. The stress pattern consists of a sequence of vowels and their stress statuses. The rhythm pattern consists of a sequence of stressed vowels and their timing information, including the start and end time stamps of each vowel (see Figure 1.2). This input is used later in Span.

There is one output — a list of identified stress or rhythm errors — from Span. The identified errors are descriptions of misplaced stressed vowels or unmatched time intervals in the speech. This output is conveyed to Peco, which uses the reported information to provide useful individualised feedback to the user.

Inside Span there are three key components. The first component is a speech recogniser, which takes the text and the sound, performs phoneme level forced alignment by using a set of pre-trained phoneme HMMs [85], and produces a phoneme labelled sound. The second component is a stress detector, which analyses the phoneme labelled sound, focusing on vowel segments to identify which vowel phonemes would be perceived as stressed by using a pre-trained classifier and produces a stress pattern of the speech. The third component is an error identifier, which examines the generated stress pattern to identify stress errors by comparing it with a target stress pattern. Then it further identifies rhythm errors if no serious errors are found in the user's stress pattern.

1.4 Issues Addressed

1.4.1 Constructing the Speech Recogniser

In order to analyse an ESL learner's speech, we first need to make our system be able to recognise the speech. We use a toolkit named Hidden Markov Model Toolkit (HTK) [85] to build our speech recogniser. HTK is a statistical-based speech recognition system. Speech is encoded using a frame-based digital signal processing method [85] and is modeled statistically using HMMs.

There are many parameters used to configure HMMs and the speech encoding process and it is important to set them to the optimal values.

HTK provides default settings for a standard speech recogniser that takes the sound waveform as the input and produces the word transcription as the output. However, this is not the kind of speech recogniser we need because we already know the exact sentence a speaker is trying to read aloud. The task of our speech recogniser is to identify the phonemic elements in a given ESL learner's utterance so that later we can determine the stress pattern of the utterance, and identify prosodic errors in the learner's speech. These later tasks — stress detection and error identification — require accurate start and end time information for each phoneme segment in an ESL learner's speech. Since the default settings for configuring phoneme HMMs and speech encoding provided by HTK may not be appropriate for our task, we must determine optimal values of the parameters in order to appropriately construct our speech recogniser so it can produce phoneme level labelled sound with as accurate as possible start and end time stamps.

1.4.2 Building the Stress Detector

In order to identify the prosodic errors in an ESL learner's speech, we must identify the stress pattern of the speech first. There are two types of stress in English — lexical stress and rhythmic stress [45]. We must determine which kind of stress pattern we should detect and analyse.

There are several prosodic features relevant to stress, including duration, amplitude, and pitch (also known as fundamental frequency). These features are easily extracted but the issue is how we should normalise them in order to reduce effects of variations due to differences such as those between speakers, recording situations, and utterance contexts. Vowel quality is known to be another important feature for identifying the stress status of a vowel [18]. The measurement of vowel quality is not trivial since there is no standard way to calculate it. So another issue is how we can measure vowel quality properly. Furthermore, among these features, we need to identify those that are most useful for classifying the stress status of a vowel as stressed or unstressed.

It is sensible to use machine learning technology to construct a classifier for detecting stress status of a vowel since the stress detection task can be considered as a binary classification problem. However, there are many machine learning algorithms, we need to find one that does a good job for this task.

1.4.3 Creating the Error Identifier

Once we have completed the speech recognition and the stress detection procedures, we will have a stress pattern and a derived rhythm pattern of the ESL learner's speech. We need to compare the generated stress and rhythm patterns with the given target patterns. However, this comparison is not trivial. To compare either the stress status or rhythm property of vowels in the ESL learner's speech with the target speech, we need to ensure the vowel pairs are properly aligned. Due to many variations and errors in an ESL learner's speech, such as adding extra phonemes, mispronouncing phonemes, or replacing phonemes by others, the vowel phoneme sequence in the ESL learner's speech is unlikely to be the same as the target. In such a situation, the issue is how we can make the system overcome ambiguities and difficulties to properly align the vowel sequences.

Once we have aligned the vowel sequences, the stress and rhythm differences can be explored. However, we need to determine what kind of differences represent errors that need to be fixed by users. In order to produce the most useful personalised feedback to users, it is important to ignore consequential or minor errors and report only critical errors to Peco.

1.4.4 Summary

The main issues covered by this thesis can be summarised as follows:

- how to configure phoneme HMMs for constructing our speech recogniser;
- how to configure the speech encoding process for constructing our speech recogniser;
- which type of stress we should work on;
- how to normalise prosodic features for detecting stress;
- how to calculate vowel quality features for detecting stress;
- which features are the most useful for detecting stress;
- which machine learning algorithm we should use to build up the stress detector;
- how to overcome ambiguities and difficulties to properly align vowel sequences;
- how to determine the critical stress and rhythm errors in ESL learners' speech.

1.5 Contributions

This thesis has the following major contributions:

1. This thesis shows how to choose a set of parameters for constructing a forced alignment speech recogniser. An exhaustive experiment that covers 3,555 combinations of parameters provides empirical evidence on choosing optimal parameter values. A client/server computing system was developed to make this experiment feasible.

   Part of this work was published in [83].

2. This thesis presents a novel method for calculating and measuring vowel quality features. Our vowel quality features are shown to achieve comparable performance to other prosodic features in automatic rhythmic stress detection.

   Part of this work was published in [82].

3. This thesis shows how to automatically identify especially prosodic stress and rhythm errors in ESL learners' speech. To our knowledge, very little research has been done on automatic stress and rhythm error detection. Our study itself provided a considerable useful example for further research in this field.

1.6 Thesis Outline

The rest of this thesis is structured as follows:

Chapter 2 starts by reviewing Automatic Speech Recognition (ASR), including techniques in the digital signal processing area, then describes suprasegmental properties of speech, including explanations of stress and rhythm. Discussions of features believed to be relevant to the perception of stress and vowel quality are also presented. It also surveys work closely related to this thesis.

Chapter 3 describes how to recognise a user's speech, which involves designing phoneme HMMs and configuring speech encoding processes, and presents an exhaustive search experiment and its results.

Chapter 4 describes how to classify vowel segments as stressed or unstressed. It also introduces a novel method for measuring vowel quality features for classifying stress. Two machine learning algorithms for constructing the stress classifier are also discussed.

Chapter 5 describes the automatic stress and rhythm error identification approach. It also presents visualisation tools for helping users understand their prosodic problems.

Chapter 6 summarises the achievements, and presents possible future research work.

Note that International Phonetic Alphabet (IPA) symbols are used throughout the text of this thesis to represent phonemes, and the New Zealand Spoken English Database (NZSED) [1] phonemic labels are used in the figures. A list of IPA symbols, NZSED labels, and example words is given in Appendix A.

[1] The NZSED is a database created by the School of Linguistics and Applied Language Studies at Victoria University of Wellington. It contains a representative sample of spoken New Zealand English from 1990s . For more information, see `http://www.vuw.ac.nz/lals/nzsed/`.

Chapter 2

Background

This chapter reviews aspects of automatic speech recognition, suprasegmental properties of speech, and machine learning technology that are relevant to the goal of this thesis.

2.1 Automatic Speech Recognition (ASR)

"Automatic speech recognition is the process by which a computer maps an acoustic speech signal to text." [34]

2.1.1 History

ASR has been investigated for many years. In the 1870s, Alexander Graham Bell worked on a machine that was expected to be able to transcribe spoken words into written text, but failed. In 1952, at Bell Laboratories, Davis, Biddulph and Balashek developed an ASR system that could recognise the digits 0 to 9 [21]. The range of accuracy was from 50% to 100%. In 1959, at MIT Lincoln Laboratories, Forgie and Forgie's ASR system could recognise 10 vowels in a small class of mono-syllabic words [27]. The accuracy was 93%. Even more impressive, the system was speaker and sex independent. In the early 1970s, the HMM approach was developed by

Lenny Baum of Princeton University and shared with contractors of the Advanced Research and Projects Agency of the United States Department of Defense (ARPA) [19]. In the 1970's, researchers around the world made several significant contributions to speech recognition area [37, 63, 67, 74]. After that, researchers gradually moved towards end-user products, signalled by the foundation of companies such as Dragon Systems in 1982 and SpeechWorks in 1984 [19].

Today, ASR has left academic labs and entered into the homes of end-users. There is now commercially available software for personal computers such as Dragon Systems' Dragon Naturally Speaking, Lernout and Hauspies Voice Xpress, and IBM's ViaVoice® [19]. These systems can perform continuous dictation with large vocabularies, and can process context-sensitive spoken commands. These systems also learn on the fly – if you correct the system when it mis-recognises a word, it will "learn". In this way, the training never actually ends, and the more you use and correct the system, the better it becomes at recognising your voice. Even though ASR has made such significant progress, it still requires further research and development to recognise continuous natural speech from multiple speakers [65].

2.1.2 Recognition Categories

Speech recognition techniques can be divided into many categories using different criteria. This section briefly discusses recognition categories based on input and output differences.

In terms of output, speech recognition systems can be divided into word level recognition and sub-word level recognition, including syllable level recognition and phoneme level recognition [85].

In terms of input and the focus of the speech recogniser, recognition can be considered to have two kinds —*full recognition* and *forced alignment*.

With "full recognition", the recogniser only has the digitised sound signal as input, and the output is the transcription of words, syllables, or phonemes. Full recognition is usually used when a system does not know but wants to find out what is being said in a given set of words, syllables, or phonemes. But the timing information of each recognised unit is not really a concern [64]. If large vocabularies are used, then one way of obtaining good performance is to use language models or artificial grammars to restrict the combination of words and reduce the search space [85].

With "forced alignment", the recogniser has at least two inputs. One is the digitised sound signal; the other is the text of the sentence that the speaker uttered [64, 68, 85]. The text can be a sequence of words, syllables, or phonemes. Forced alignment is usually used to segment the sound signal and annotate it with the elements in the text. In contrast to full recognition, the exact sequence of spoken words is known; the goal of the recogniser is to identify the start and end time points of each segment unit [64, 68]. So researchers can use the timing information for other purposes [25, 58, 71].

If a system focuses on getting high accuracy on timing information of individual segments, then using forced alignment is generally a good choice provided the content of sentences is given.

2.1.3 Digital Signal Processing

According to [48], although in theory the raw digitised sound waveform may be used directly in speech recognition, the unlimited variability of sound signal makes this infeasible. Thus researchers tend to use a limited set of high level extracted features to represent the raw digitised sound signal. Feature vectors are typically extracted every 10–20 ms frame period using a window size of 20–30 ms [48]. So a raw digitised sound signal can be characterised by a sequence of feature vectors. The most frequently used feature extraction techniques are: linear predictive coding (LPC) [49], Mel-frequency cepstral coefficients (MFCC) [22], and perceptual linear prediction (PLP) [32].

LPC

LPC starts with the assumption that the speech signal is produced by a buzzer at the end of a tube. The glottis produces the buzz, which is characterised by its intensity and frequency. The vocal tract forms the tube, which is characterised by its resonances, called formants. LPC preforms a three step speech signal analysis, which includes estimating formants, removing their effects from the speech signal, and estimating the intensity and frequency of the remaining buzz. The process of removing the formants is called inverse filtering, and the remaining signal is called the residue [33].

LPC is one of the most powerful techniques used to represent a speech signal. Usually it is implemented as an all-pole model to capture the vocal tract properties. It is the commonly used method for encoding clean speech at a low bit rate. However its performance degrades if a speech signal contains distortions [76]. Another drawback is that amplitude seems to be a very important feature to differentiate nasalised consonants and voiced vowels, but it is not included in the standard LPC formant analysis procedure [69].

MFCC

MFCC is based on the known variation of the human ear's critical bandwidths with frequency: filters spaced linearly at low frequencies and logarithmically at high frequencies capture the phonetically important characteristics of speech. This is expressed in the Mel-frequency scale, which is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz.

Figure 2.1 illustrates the Mel-scale filter bank.

MFCC is one of the standard representations used by HTK [85], a

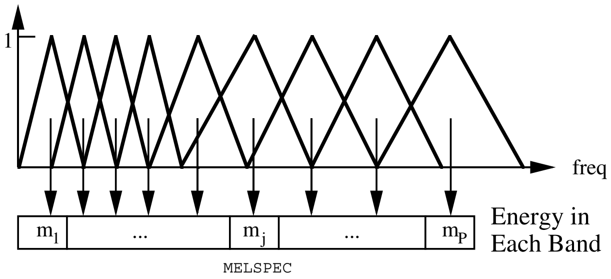


Figure 2.1: Mel-scale filter bank (adapted from [85]).

toolkit involved in our system. A conventional method for extracting the mel-frequency cepstral features includes the following steps:

- Window the data with a Hamming window
- Take the fast Fourier transform
- Find the magnitudes of the output of the fast Fourier transform
- Convert the data into filter bank outputs
- Calculate the decimal logarithm
- Find the discrete cosine transform

Note that the purpose of performing the cosine transform at the last step is to decorrelate the set of log energies to a set of uncorrelated cepstral coefficients [35]. Thus only about half of the cepstral coefficients — typically 12, from 1st to 12th [85] — will be left for use in a further speech recognition process. The 0th cepstral coefficient describes the shape of the log spectrum independent of its overall level so that it can be used to estimate the energy over a short period. The 1st cepstral coefficient represents the balance between the upper and lower halves of the spectrum. The higher order coefficients are related to increasingly finer features in the spectrum [15].

Compared with the large amount of Mel-Scale bands, the smaller size cepstral coefficients make it easier to compute reasonably accurate probability estimates in a subsequent statistical recognition process.

PLP

The PLP technique may be thought as the combination of LPC and MFCC. Its features smooth an auditory spectrum by fitting an auto-regressive model. The auto-regressive model can be an all-pole model, which is the model mostly used in LPC, or a linear-predication model. The features are computed based on the Bark-scale [86], which is similar to the Mel-scale in modelling the human ear.

According to [24], "PLP has the nice property of modeling spectral peaks more carefully than the less-reliable valleys in between." It generally outperforms LPC [73] but not MFCC [24].

2.1.4 Hidden Markov Models

There are several kinds of approaches used in ASR, including the template-based approach, the knowledge-based approach, the statistical-based approach, and the connectionist-based approach [65, 70]. We will use one example of the statistical-based approach — Hidden Markov Models — in this research. This section briefly introduces the HMM technique.

From the statistical point of view, the speech recognition problem is the problem of computing the probability that a vocabulary word is spoken within an utterance. This probability is not directly computable but can be indirectly computed once the likelihood of each HMM generating the sound of that word has been worked out [85].

The success of HMM based speech recognition is built on an assumption that the sequence of observed speech feature vectors corresponding to each word is generated by a Markov model. A Markov model can be viewed as a finite state machine. When a state is entered, a speech feature vector is generated from a probability density. When all speech feature vectors are generated, the likelihood of the model generating the feature vectors can be calculated. *Hidden Markov Model* is named because we just know the sequence of speech feature vectors but do not know what exactly the underlying sequence of states are [85].

In order to obtain good speech recognition performance, one of the keys is to get a well trained model for each recognised unit, such as word, syllable, or phoneme. Providing a sufficient amount of training data is essential for obtaining well estimated models. However, we think that the HMM design itself and an accurate speech signal encoding process are more important. We hypothesise that:

Hypothesis 2.1.1 *Without careful consideration of parameters involved in the HMM design and the speech encoding process, the speech recognition performance would be low.*

2.2 Suprasegmental Properties

Suprasegmental properties of speech are properties that cannot be derived directly from the underlying sequence of phonemes. Researchers divide suprasegmental properties into several categories and the most commonly used terms for them are stress, rhythm, tone and intonation [4, 11, 17, 18, 30, 43, 50].

Suprasegmental properties are very important for the comprehensibility of speech. ESL speakers with excellent vocabulary, grammar, and individual phoneme pronunciation may still be very hard to understand if the stress, rhythm, tone, or intonation patterns of their speech are wrong. In this thesis, we only focus on two suprasegmental properties — stress and rhythm.

2.2.1 Stress

Stress is a form of prominence in spoken language. Usually, stress is seen as a property of a syllable or of the vowel nucleus of that syllable. There are two types of stress in English [45]. *Lexical stress* refers to the relative prominences of syllables in individual words. *Rhythmic stress* refers to the relative prominences of syllables in longer stretches of speech than isolated words. When words are used in utterances, their lexical stress may be altered to reflect the rhythmic (as well as semantic) structure of the utterance.

Lexical stress is usually used in the discrete speech environment and can be thought of as a syllable's potential to receive prominence. In English, there are many noun-verb phonetically similar word pairs with different lexical stress patterns. For instance, the word "project", when it is used as a noun, the lexical stress is placed on the first syllable, but when it is used as a verb, the lexical stress is placed on the second syllable. Incorrect lexical stress placements can often cause grammatical problems, such as a listener expecting a noun but hearing a verb.

Rhythmic stress is usually used in the continuous speech environment and can be thought of as the actual degree of prominence observed when a syllable is uttered as part of a sentence.

Lexical stress and rhythmic stress often coincide, but due to the occurrence of stress shifting, they may differ in continuous speech. For example, in the sentence "The total number of people is thirteen", the lexical and rhythmic stressed syllables of the word "thirteen" are the same — the second syllable. But in the sentence "There are thirteen people in this room", the rhythmic stressed syllable of the word "thirteen" is often the first syllable, which is not the same as the lexical stress.

2.2.2 Rhythm

Speech rhythm is defined as the sequence of the durations of consecutive vowels, consonants, and pauses in speech [14]. As introduced in [1, 60], speech rhythm comes in two types: *stress-timed rhythm*, where the stresses recur at approximately equal time intervals, and *syllable-timed rhythm*, where the syllables recur at regular intervals. A widely held theory is that speech rhythm in English is stress-timed [1, 6, 17, 18, 20, 43, 56, 61]:

that is, the intervals between pairs of stressed syllables in an English sentence are approximately equal. Although generally accepted, the literature [10, 18, 20, 43, 66] notes that this theory has not yet been experimentally verified.

2.2.3 Perceptual Studies of Features for Stress

The perception of a syllable as stressed or unstressed depends on its relative duration, its amplitude or energy, its pitch, and its quality, especially the *vowel quality* [18]. Many researchers have tried to determine which feature has the most significant effect in signaling stress in speech. However, the current results in the literature are inconclusive.

Fry [29] investigated the effects on stress perception by changing syllable pitch, energy and duration. The data set consisted of noun–verb bi–syllable words from synthetic utterances. Fry concluded that for lexical stress perception the most important cue was pitch.

Lieberman [46] examined the relationship between the perceived syllable stress and acoustic correlates, including peak envelope amplitude, syllable duration, and pitch. The data set consisted of noun–verb bi–syllabic words extracted from natural speech. Lieberman indicated that envelope amplitudes and pitch were the most relevant unidimensional cues of stressed syllables and amplitude was more important than pitch.

Adams and Munro [2] explored the use of pitch, amplitude, and duration of syllable stress perception in longer utterances. Pitch contour of the syllables, amplitude envelope of the syllables, and duration of the whole syllable were extracted from longer utterances. They claimed that a greater degree of pitch change, a greater fall of amplitude from a fairly constant peak level, and longer duration were linked to syllable stress perception. They suggested that syllable duration is the most frequently used cue and syllable amplitude is the least used.

2.2.4 Vowel Quality

As just mentioned at the beginning of the previous sub-section, a further correlate of stress is the quality of the vowel in a syllable. According to the traditional theory, vowel quality is determined by the configuration of the tongue, jaw, and lips [7, 42, 44, 59], so that there are three main parameters describing vowel quality. These three parameters form a three-dimensional vowel space: the position of the highest point of the tongue in the close-open dimension, and in the front-back dimension; and the degree of lip rounding in the rounded-spread dimension. Therefore, vowel quality can be represented by a X-Y-Z form, where X refers to the horizontal axis of the tongue position and includes front, central and back, Y refers to the vertical axis of the tongue position and includes high/close, low/open, and mid (it can also be divided into half-close and half-open), and Z refers to the degrees of lip rounding and includes rounded and unrounded. A three-dimensional vowel space is shown in Figure 2.2. However, these three parameters cannot be measured from the sound. Other correlates that can be calculated from the sound are needed to describe vowel quality.

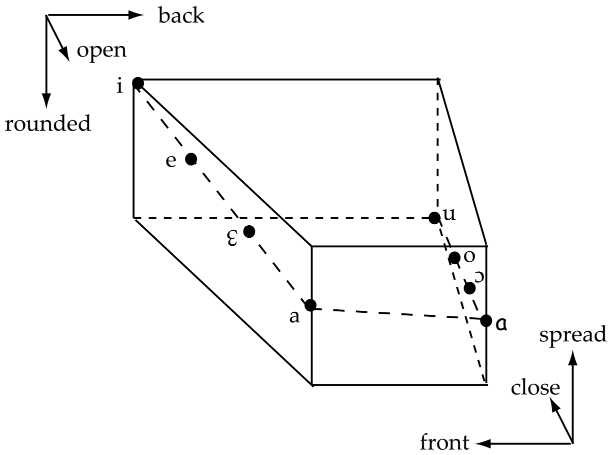


Figure 2.2: A 3D model of the vowel space (adapted from [42]).

Ladefoged [43] showed that the height of the tongue correlates to the frequency of the first formant (F_1) and the backness of the tongue correlates to the difference between the frequencies of F_1 and F_2 (the second formant), but there does not exist an auditory property correlating to the rounding of the lip. Ladefoged [43] also mentioned that height and backness were the mostly used features to distinguish one vowel from another in nearly every language, including English. Therefore, as height and backness correlate to formants and formants can be easily calculated directly from sound signal, the three-dimensional vowel space can be reduced to a two-dimensional vowel space. A simple two-dimensional vowel space is shown in Figure 2.3.

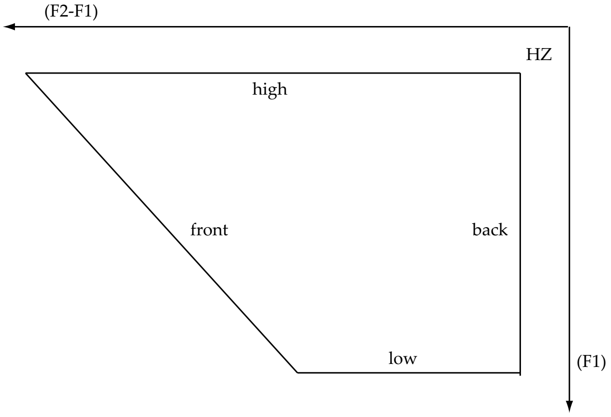


Figure 2.3: A simple 2D vowel space (adapted from [43]).

However, we realised that the first two formants are not sufficient to identify quality of different vowels due to flexibility in the formation of a vowel and variation of speakers. Figure 2.4 illustrates the vowels (excluding diphthongs) in F_1 and F_2 space for the female New Zealand (NZ) native speakers (left) and the male NZ native speakers. The centroid of each vowel class is mark by the lexical item and the perimeter of each ellipse indicates the boundary of the area in which at least 95% of the data for a particular class fall. From Figure 2.4, we can see that overlaps among these vowels casue the problem of identifying vowel qualities amongst multiple speakers' speech by using F_1 and F_2.

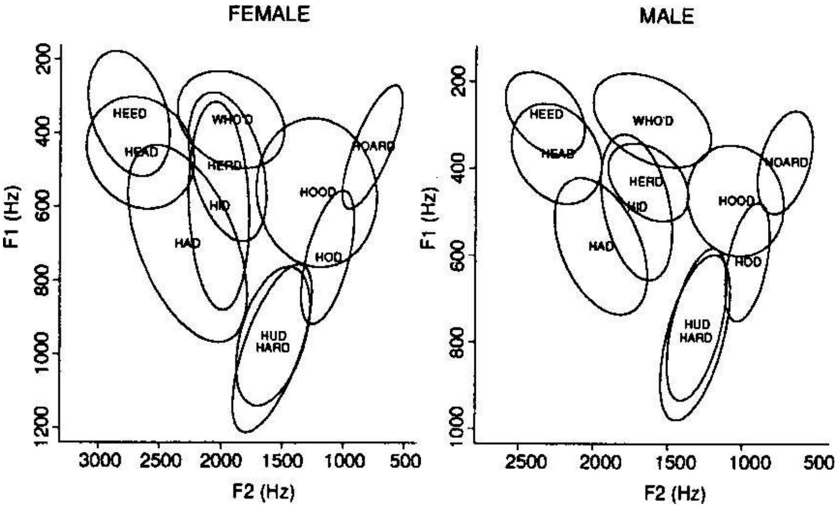


Figure 2.4: Vowels in F1/F2 space (from [78]).

Cruttenden [18] further reduced the dimensions describing vowel quality to one dimension. He introduced the notions of full or reduced vowel to describe vowel quality for detecting stressed syllables. However, there is no standard method for determining whether a pronounced vowel is full or reduced. The term "full vowel" and "reduced vowel" abstractly describe the quality of a vowel. In English phonology, "reduced" means among other things, central in articulatory vowel space, short and unstressed [18, 43]. In contrast to "reduced", "full" vowel means more peripheral in vowel space. If the quality of a vowel is considered to be "reduced", then the vowel is always unstressed. However, even if a vowel is considered to be "full", the vowel may occur in both stressed or unstressed syllables [43].

We are interested in using the one-dimensional vowel space to detect vowel stress status. Based on the above discussion, we hypothesise and will show that:

Hypothesis 2.2.1 *For detecting stress status, knowing a vowel is reduced is more reliable than knowing it is full.*

2.3 Machine Learning

Machine learning is one of the hottest research areas. It has been widely adopted in real-world applications, including speech recognition, handwritten character recognition, image classification and bioinformatics. This section gives a brief overview of the machine learning technology related to this thesis.

2.3.1 Definitions

Researchers give different definitions of machine learning. However the principle is roughly the same: a computer program processes a given set of examples and tries to either describe the known data in some meaningful ways or develop an appropriate response to unseen cases. We list three of representative definitions or descriptions as follows:

Mitchell [52] gives the following definition of machine learning:

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improve with experience E."

Witten and Frank [81] state that:

> "... things learn when they change their behavior in a way that makes them perform better in the future ..."

Michalski et al. [51] state that:

> "Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time."

2.3.2 Terminology

A data set is a collection of knowledge or examples. A single example in a data set is called an *instance*. There are one or more *attributes* or *features* representing the aspect(s) of an instance. Each attribute or feature can have either a categorical or numerical value.

In order to train a computer program and evaluate its performance, the data set is usually split into two subsets: *training data set* and *test data set*. We sometimes split the data set into three subsets. The third data set is usually called *validation data set*. The purpose of using validation data set is to monitor the training progress and prevent the training from overfitting. When no extra data are available to provide a separated validation data set, the *n-fold cross validation* method is sometimes used to overcome the overfitting problem [52]. The m available examples are randomly partitioned into n disjoint subsets, each of size m/n. Training and validation processes are then run n times. In each run, a different one of these n subsets is used as the validation data set and all the other subsets are merged and used as the training data set. The averaged validation result is used to evaluate the training performance.

2.3.3 Learning Paradigms

According to [38], based on the knowledge provided, there are three main learning paradigms, *supervised learning*, *unsupervised learning*, and *hybrid learning*. Supervised learning is sometimes referred to as learning with a teacher. The knowledge provided to a learning system includes a correct answer for each input instance. The learning process is continued until the learning system produces answers as close as possible to the given correct answers. Unsupervised learning is sometimes referred to as learning without a teacher. Instances are grouped into appropriate categories by analysis. A typical problem dealt with by unsupervised learning is *clustering* [26, 31]. "Hybrid learning combines both supervised and unsupervised learning. Part of the solutions (network weights, architecture, or computer programs) are determined through supervised learning, while the others are obtained through unsupervised learning." [38]

There are many machine learning models in common use. Here we briefly describe some of them:

- **Decision Trees.** A decision tree consists of leafs and nodes. A leaf records an answer (often called a *class*) and a node specifies some test conditions to be carried out on a single feature value of an instance, with one branch and sub-tree for each possible result of the test. For a given instance, a decision is made by starting from the root of a tree and moving through the tree determined by the outcome of a condition test at each node until a leaf is encountered. [62]

- **Neural Networks.** A neural network is usually constructed from nodes, links, weights, biases, and transfer function. Nodes are often called *neurons*. There are varieties of ways to connect neurons. Usually multiple layers are needed to manage the neurons. Through network training the internal weights and biases of the neurons are automatically updated in order to produce the target output. [54]

- **Support Vector Machines.** In a support vector machine, input vectors are mapped into a very high-dimension feature space through a non-linear mapping. Then a linear classification decision surface is constructed in the high-dimension feature space. This linear decision surface can take a non-linear form when it is mapped back into the original feature space. Special properties of the decision surface ensures good generalisation ability of this learning machine. [16]

- **Genetic Programming.** Genetic programming has become more and more important in the machine learning research area since the 1990s. The inputs to a genetic programming learning system are a terminal set and a function set. The outputs of the system are evolved computer programs. A fitness function is used to evaluate individual programs during the learning in order to select winners, then several evolutionary operations are applied to these winners to form the next generation, and so on in order to produce the best individual program for a given problem. [41]

2.4 Related Work

2.4.1 Speech Recognition

There is vast literature on ASR. We briefly review some of the articles, which are closely related to HMM-based systems and contain information on parameters used in HMM design and speech encoding.

Bocchieri et al. [8] constructed a speech recogniser by using triphone HMMs. The speech signal was encoded by using a 20 ms window and shifted by a 10 ms interval. The encoded speech frame feature vector consisted of 12 MFCCs and the frame energy measured in dB with their first and second derivatives, making 39 components in total. However there were no clear description of how many states the HMM had and how these states linked. Gaussian mixtures [85] were used to define the continuous state observation densities but the number of Gaussian mixtures was not stated. The recogniser had 3.6% word error rate.

Rapp [64] constructed a forced alignment system for German. The system used 25.6 ms window size with 10 ms interval to calculate sound feature vectors, which consisted of 12 MFCC features and overall energy, and their first and second derivatives. A single mixture Gaussian output probability density function was used in the phoneme HMMs. The author explored three different phoneme HMM topologies and reported that by using 3-state left-to-right HMM topology, the percentage of auto-labelled phoneme boundaries within a 20 ms threshold from the manual labelling was 84.4%.

Wightman and Talkin [79] developed an HMM-based forced alignment system with acoustic model training and Viterbi search [85] implemented using the HTK. In their system, the frame period was 10 ms and each HMM state contained five Gaussian mixtures. The TIMIT database[1] was used and approximately 80% accuracy within the 20 ms threshold was reached.

Pellom and Hansen [57] developed a gender-dependent forced alignment system by using 5-state left-to-right HMMs. For each HMM state, 16 Gaussian mixtures were used. The speech waveform was parameterised every 5 ms by a vector consisting of 12 MFCCs, 12 delta MFCCs and normalised log-frame energy. However, no clear statement about the window size was found. The TIMIT database was used and the phoneme segmentation accuracy was 85.9% within the 20 ms threshold.

Pellom [58] developed a 5-state left-to-right HMM-based forced alignment system with an output distribution of 16 Gaussian mixture densities per state. By using 25 ms window size with 5 ms frame interval, 12 MFCC features and the normalised log frame energy plus their first derivatives were calculated. The phoneme segmentation accuracy within 20 ms threshold was 86.2%.

Irino et al. [36] presented a new speech recognition/generation system. The system consisted of STRAIGHT [40], warped-frequency discrete cosine transform, and an HMM engine. Each HMM has 12 states with 2 Gaussian distributions per state. MFCC features were computed over 40 ms window with 2 ms interval. The orders of the MFCC features adopted in the experiments were 12, 20 and 30. No first and second derivatives were used since the authors intended to compare the potential of MFCC baseline features with others. Their experiments reported that the speaker-dependent word recognition rates were between about 95 and 97%.

Lindgren et al. [47] introduced a novel method for speech recognition.

[1] One of the Linguistic Data Consortium (LDC) top ten corpora. See `http://www.ldc.upenn.edu/Catalog/topten.jsp`

In order to evaluate the new method, the authors constructed a HMM-based speech recogniser by using a set of one-state HMMs with 16 Gaussian mixtures and used it in a performance comparison experiment. In the HMM-based baseline speech recogniser, the features used were 12 MFCC, log energy and their deltas and delta-deltas. However, no clear statements about the window size and frame period were found. The accuracy for isolated phoneme recognition was 54.86%.

The review above indicates that there are no standard settings for those parameters in constructing either a speech recogniser, a forced alignment system, or other similar systems. It seems that these parameters need to be adjusted for different purposes using different speech data sets in different situations. In order to obtain optimal performance, careful analysis and further investigation in these parameters are essential.

2.4.2 Stress Detection

There have been a number of reports on stress detection. Most reports focused on lexical stress detection based on isolated words, but a few have addressed rhythmic stress detection in complete utterances.

Lieberman [46] used duration, energy and pitch to identify lexical stress in bisyllabic noun-verb stress pairs (e.g. PREsent vs preSENT). These features were extracted from the hand-labelled syllables. The database consisted of isolated words pronounced individually by 16 native speakers and was used for both training and testing. A decision tree approach was used to build the stress detector and 99.2% accuracy was achieved.

Aull and Zue [5] used duration, pitch, energy, and changes in the spectral envelope (a measure of vowel quality) to identify lexical stress in polysyllabic words. The features were extracted from the sonorant portion of automatically labelled syllables. The pitch parameter used was the maximum value in the syllable. Energy was normalised using the logarithm of the average energy value. The database consisted of isolated words extracted from continuous speech pronounced by 11 speakers. A template-based algorithm was used to build the stress detector and 87% accuracy was achieved.

Ferij et al. [28] used pitch, energy and the spectral envelope to identify lexical stress in bi-syllabic words pairs. The first and second derivatives of pitch and the first derivative of energy were also used. The spectral envelope was represented by four LPC features. These nine features were extracted from the hand-labelled syllables at 10 ms intervals. The database consisted of isolated words extracted from continuous speech pronounced by three male speakers. HMMs were used to build the stress classifier and an overall accuracy of 94% was achieved. Vowel quality, especially the distinction between reduced and full unstressed syllables was suggested as a direction for future work.

Ying et al. [84] used energy and duration to identify stress in bi-syllabic word pairs. Energy and duration features were extracted from automatically labelled syllables and were normalised using several methods. The database consisted of isolated words extracted from continuous speech pronounced by five speakers. A Bayesian classifier assuming multivariate Gaussian distributions was adopted and the highest performance was 97.7% accuracy.

A few studies have investigated stress detection in longer utterances. Waibel [75] used amplitude, duration, pitch, and spectral change to identify rhythmically stressed syllables. The features were extracted from automatically labelled syllables. Peak-to-peak amplitude was normalised over the sonorant portion of the syllable. Duration was calculated as the interval between the onsets of the nuclei of adjacent syllables. The pitch parameter used was the maximum value of each syllable nucleus. Spectral change was normalised over the sonorant portion of the syllable. The database consisted of 50 sentences read by 10 speakers. A Bayesian classifier assuming multivariate Gaussian distributions was adopted and 85.6% accuracy was reached.

Jenkin and Scordilis [39] used duration, energy, amplitude, and pitch to classify vowels into three levels of stress — primary, secondary, and unstressed. The features were extracted from hand-labelled vowel segments. Peak-to-peak amplitude, energy and pitch were normalised over the vowel segments of the syllable. In addition syllable duration, vowel duration and the maximum pitch in the vowel were used without normalisation. The database consisted of 288 utterances (8 sentences spoken by 12 female and 24 male speakers) from dialect 1 of the TIMIT speech database. Neural networks, Markov chains, and rule-based approaches were adopted. The best overall performances ranged from 81% to 84% by using Neural networks. Rule-based systems performed more poorly, with scores from 67% to 75%.

Van Kuijk and Boves [72] used duration, energy, and spectral tilt to identify rhythmically stressed vowels in Dutch — a language with similar stress patterns to those of English. The features were extracted from manually checked automatically labelled vowel segments. Duration was normalised using the average phoneme duration in the utterance, to reduce speaking rate effects. Also a complex duration normalisation method introduced in [80] was adopted. Energy was normalised using several procedures, such as the comparison of the energy of a vowel to its left neighbour and its right neighbour, to the average energy of all vowels to its left and to the average energy of all vowels in the utterance. Spectral tilt was calculated using spectral energy in various frequency sub-bands. The database consisted of 5000 training utterances and 5000 test utterances from the Dutch POLYPHONE corpus [23]. A simple Bayesian classifier was adopted, on the argument that the features can be jointly modelled by a N-dimensional normal distribution. The best overall performance achieved was 68%.

The summary above shows that stress classification has higher accuracy for a limited task, such as identifying the stressed syllable in isolated bi- or polysyllabic words, but performance levels are noticeably lower in the few studies using longer utterances. Vowel quality was addressed in few studies. A variety of machine learning techniques were used in the studies but it does not seem to indicate that a particular classification procedure is any more successful than any other.

2.4.3 Prosodic Error Identification

We have not yet found any literature relevant to the automatic prosodic error identification issue.

2.5 Chapter Summary

In this chapter, we have briefly reviewed ASR and related technologies, including signal processing and HMMs. We have discussed stress and rhythm properties of speech and explained vowel quality in terms of stress detection. We have also presented closely related work to our study. From the next chapter, we will present our work in each of the three stages in Span.

Chapter 3

Speech Recognition Stage

This chapter discusses the speech recognition stage as highlighted in Figure 3.1. Section 3.1 gives a more detailed overview of the speech recognition stage, including the procedure for training phoneme HMMs. Sections 3.2 and 3.3 discuss issues in phoneme level HMM design and speech encoding. Section 3.4 describes an experiment for exploring sets of parameters for building an effective speech recogniser. Section 3.5 presents the experimental results. Section 3.6 summarises this chapter. Note that part of this chapter is taken from [83].

3.1 Overview

The speech recognition stage is the first stage of Span. It is implemented using the procedures in HTK. The performance element of this stage is an HMM-based speech recogniser. We trained phoneme HMMs from a speech data set, hand labelled at the phoneme level. As introduced in section 1.4.1 (page 5), our speech recogniser is not a standard full speech recogniser as it knows the sentence that a user is trying to say. Rather, it is a forced alignment system. That is, it represents the sentence at phoneme level and aligns the sentence with the speech signal to identify the start and end times of all the segmental units. Therefore, the input is the text of

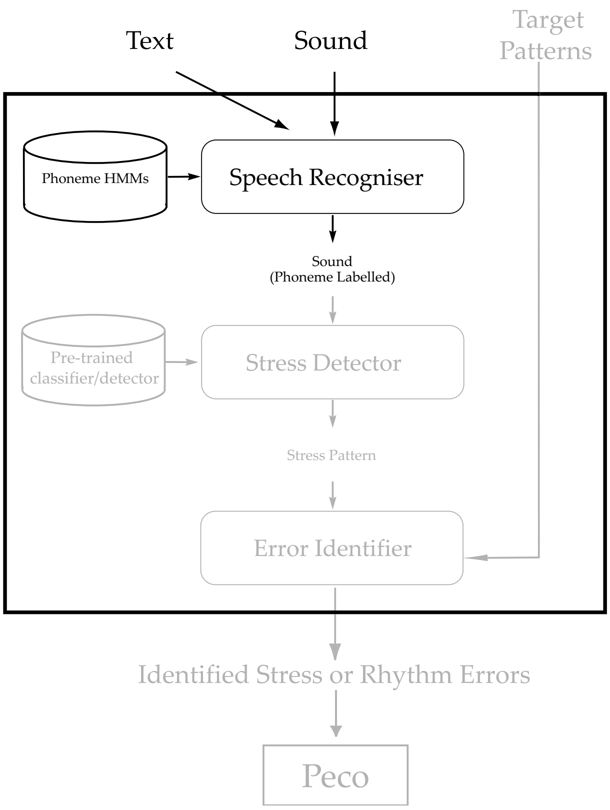


Figure 3.1: Overview of Span: recognising speech

a sentence and the sound from a user, and the output is the user's sound labelled at phoneme level.

This stage is very important because the accuracy of phoneme boundaries is critical to the later stages.

3.1.1 The Training Element

The training procedure of the phoneme level HMMs is illustrated in Figure 3.2.

The input to the training element consists of two sets of related data. One is a sound data set and the other is a phoneme label data set. These data sets are collected from NZSED.

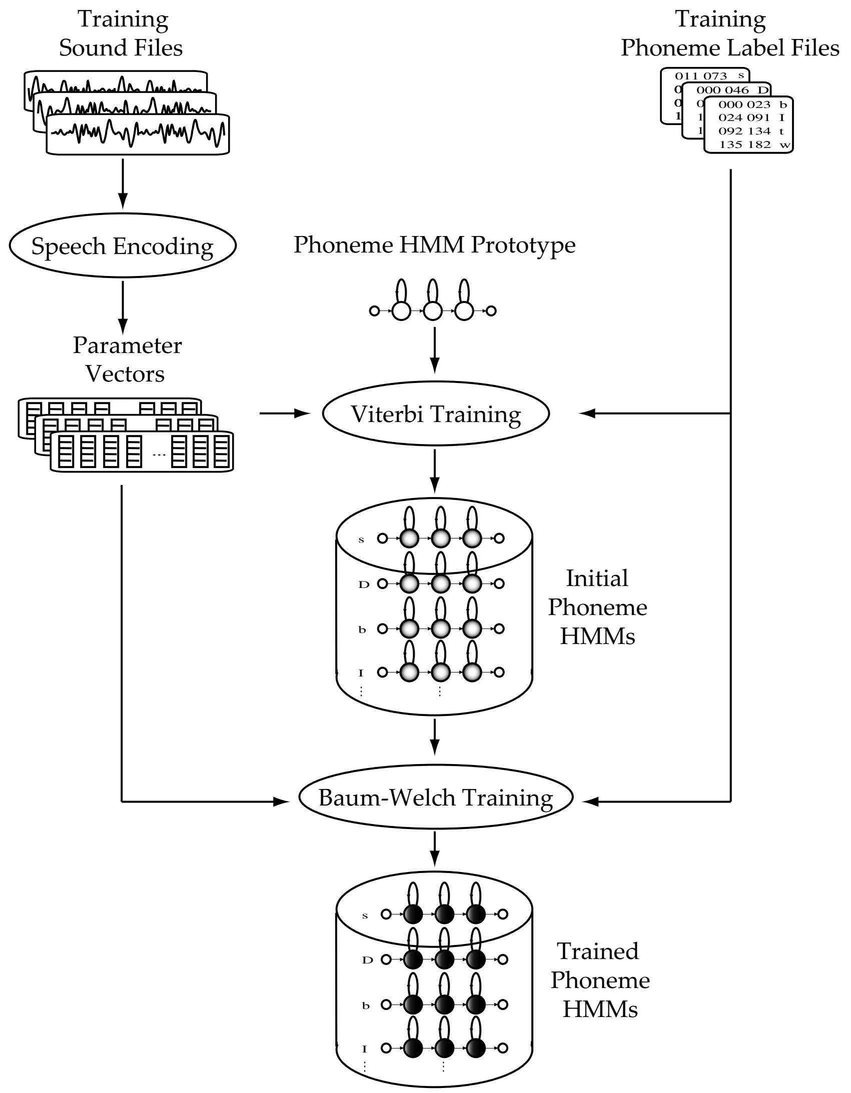


Figure 3.2: The training element.

The sound data set contains 1119 utterances of 200 distinct English sentences produced by six female native speakers. The sounds were recorded at a 16 kHz sampling rate, which allows accurate analysis of all frequencies up to 8 kHz (the Nyqvist frequency for this sampling rate). The range to 8 kHz includes all perceptually relevant information in human speech.

These sound files are then encoded into parameter vectors.

The phoneme label data set contains phoneme files corresponding to the sounds, specifying the name and the start and end times of each phoneme. The labelling was at a coarse level, with 44 different English phonemes and two labels for silences and short pauses. The first half of the 1119 utterances were hand-labelled at the phoneme level by trained linguists; the second half were automatically labelled by a computer program (based on the labelling of the first half) then checked by a trained linguist.

There are two stages in the training. The first stage is called Viterbi training. It takes phoneme HMM prototypes, which are just descriptions of the structure of phoneme HMMs, and constructs a set of initial phoneme HMMs. The second stage is called Baum-Welch training. By applying several iterations of Baum-Welch training, the initialised HMMs are gradually refined to produce phoneme level trained HMMs. More information about the Viterbi and Baum-Welch training procedures can be found in [85].

3.1.2 The Performance Element

The phoneme level forced alignment performance procedure is illustrated in Figure 3.3.

A user reads a sentence prompted on a computer screen. When the user finishes speaking, the recorded digitised sound and the text of the sentence are fed into the speech recogniser.

The speech recogniser converts the words in the text to a phoneme network. The phoneme network could be just a simple phoneme sequence if the word pronunciation dictionary includes only one pronunciation for each word. The speech recogniser also windows the sound into a sequence of short frames and encodes each sound frame into a parameter vector. It then uses a Viterbi search algorithm to calculate the acoustic likelihood for each encoded sound frame using the pre-trained phoneme HMMs and the phoneme network, and maps the sound signal with the best chosen phoneme sequence from the phoneme network. The sound labelled at the phoneme level is then output by the speech recogniser.

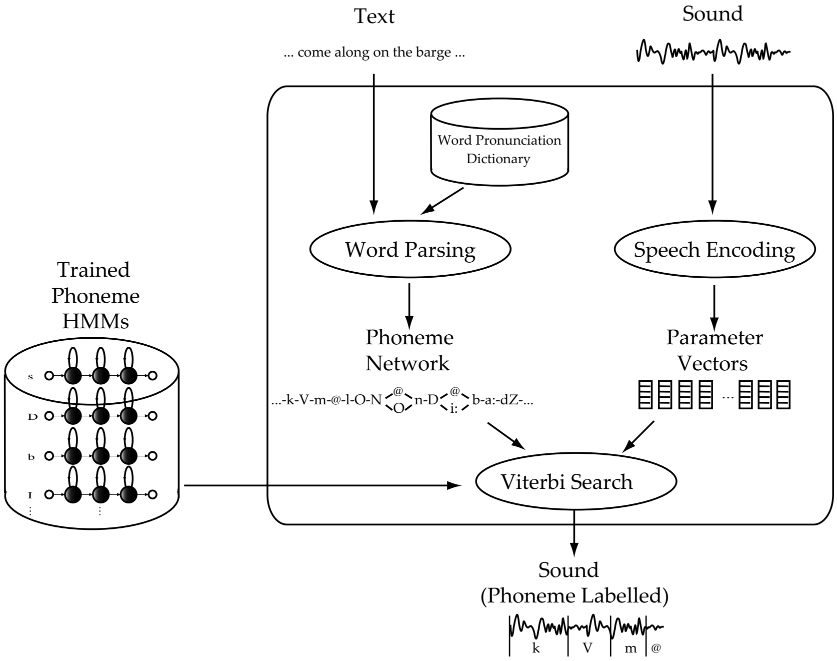


Figure 3.3: The performance element.

3.2 HMM Design

There are three key parameters required to specify the phoneme HMM prototype in the training element: the number of states needed in each phoneme level HMM, the connections between the states, and the size of the mixture-of-Gaussian models in each state of each HMM. The first two issues belong to the phoneme HMM architecture design. The last one belongs to the phoneme HMM stochastic model design.

3.2.1 Architecture of the HMMs

A phoneme-level HMM is a description of segments of input speech signals that correspond to a particular phoneme. The HMM consists of a network of states where each state describes subsegments of input speech signals that correspond to a different section of a phoneme (for example, the initial component of the phoneme).

With fewer states, an HMM will have a less precise model because each state must describe a larger section of a phoneme. It will also be less accurate at identifying the start and end times of the phoneme. With more states, the HMM may have greater accuracy, but the computational cost will be higher when recognising the speech input. It will also require more training data in order to determine all the values in the state descriptions.

To balance the need for accuracy against the computation time and size of training data, we chose to follow standard practice using a three state model for each phoneme HMM. The first state describes the transition from the previous phoneme into the current phoneme, the second state describes the middle section of the phoneme, and the third state describes the transition out of the current phoneme to the next phoneme.

The states can be connected in many ways. We chose the commonly used mode of a chained connection [85], where each state is connected to itself (since each state may cover a number of samples from the input signal) and to the next state. This mode does not allow connections that skip over a state or connect back to a previous state. An example of this connection mode is shown in Figure 3.4.

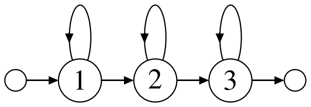


Figure 3.4: A chained connection mode HMM (adapted from [85]).

In addition to the phonemes, there are also a number of silences and short pauses in each speech signal. Because silences and pauses do not have the same regular structure as phonemes, we allowed more flexible structures for the silence and short pause HMMs: we used a modified three-state HMM with backward and forward skipping connections to model the silences and a tied one-state connection to model the short pauses, as shown in Figure 3.5.

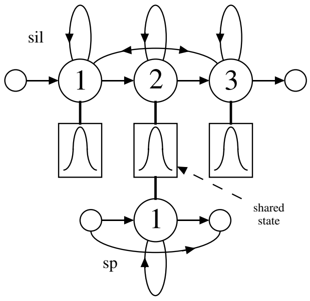


Figure 3.5: HMMs for silences and short pauses (adapted from [85]).

3.2.2 Stochastic Models: Mixture-of-Gaussians

An HMM state describes segments of speech signals using Gaussian models of each of the features used to encode the signal. If all speakers always pronounced a given phoneme in very similar ways, then there would be little variability in the signal and simple Gaussian models (mean and variance) would be sufficient. However, there is considerable variability in the usual case, and a mixture-of-Gaussians model may provide a more accurate description. The design issue is to determine an appropriate number of Gaussians in the mixture-of-Gaussian models. In general, the greater the size of the mixture-of-Gaussian model, the more training data is required to learn the parameters of the model. Particularly for those rarely occurring or unevenly distributed phonemes, providing more speech data is essential.

For our data set, we explored a range of possible sizes of the models from 1 to 16.

3.3 Speech Encoding

The HMM-based speech recogniser requires digitised sound signals to be encoded into a sequence of feature vectors, where each feature vector encodes the essential features of a short "frame" or "window" of the input signal. There are three key parameters required to configure the encoding process: the size of each window, the interval ("frame period") between two adjacent frames, and the set of features to be extracted from each frame. An encoding process is shown in Figure 3.6.

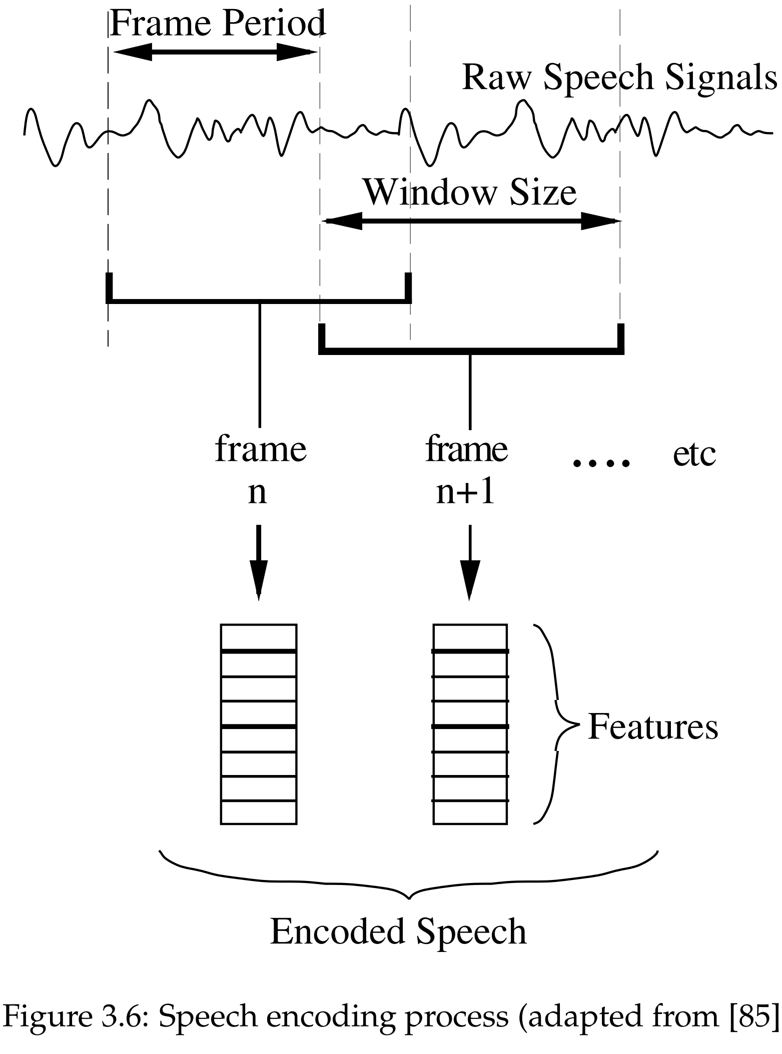


Figure 3.6: Speech encoding process (adapted from [85]).

3.3.1 Window Size and Frame Period

The window size and frame period are important parameters for the speech recogniser.

If the window size is too small, the window will not contain enough of the signal to be able to measure the desired features; if the window size is too large, the feature values will be averaged over too much of the input signal, and will lose precision. We explored a range of window sizes, from the lower limit 10 ms to the upper limit 30 ms, with the lower limit being chosen to be large enough to include at least two complete cycles of the fundamental frequency of the speech signal, and the upper limit chosen to ensure that a window seldom spanned more than a single phoneme.

If the frame period is too long, there will be insufficient feature vectors for each phoneme, which will prevent the HMMs from recognising the speech. If the frame period is longer than the window size, then some parts of the speech signal will not be encoded at all. If the frame period is too short, then there will be too many feature vectors, which will increase the computational cost. The absolute lower limit on the frame period is governed by the sample rate of the raw speech signal. We explored a range of frame periods, from 4 ms to 12 ms, subject to the constraint that the frame period was not larger than the window size.

3.3.2 MFCC Feature Extraction and Selection

There are many possible classes of features that could be used to encode a speech signal. We have followed common practice in using Mel-Frequency Cepstrum Coefficients (MFCCs) on a Hamming window. MFCCs use a mathematical transformation called the cepstrum which computes the inverse Fourier transform of the log-spectrum of the speech signal [85]. Within this class of features, there is still considerable choice about which MFCC features to use. In addition to the 12 basic MFCC transformation coefficients, there are also the energy (E), the 0th cepstral coefficient (O), and the first and second order derivatives (D and A) of those coefficients. Not all combinations of these features are sensible. For example, it makes little sense to use the second order derivatives without the first order derivatives, and the HTK toolkit [85] will not use both the energy and the 0th cepstral coefficient simultaneously. We have identified nine combinations to explore in our experiments, as shown in Table 3.1.

Table 3.1: Nine feature combinations and the number of features in each set.

No	Combination	No. of Features
1	MFCC	12
2	MFCC-D	24
3	MFCC-D-A	36
4	MFCC-E	13
5	MFCC-E-D	26
6	MFCC-E-D-A	39
7	MFCC-O	13
8	MFCC-O-D	26
9	MFCC-O-D-A	39

3.4 Parameter Identification

We decided that the architecture of phoneme HMM — the number of states and their connections — follows the commonly used one. We also decided to use MFCC features to be the main sound signal encoding representation. This section describes an exhaustive search experiment that we used to investigate the other parameters.

For our experiments, we trained a collection of phoneme level HMM models on a training set of annotated speech samples with each of combination of parameters. We then evaluated the quality of the HMM models by using them to recognise and label speech samples in a separate test set. The following sections describe the data sets used in the experiment, the parameter combinations we explored, the training and testing process, the experiment configuration, and the performance evaluation.

3.4.1 Data Set

The experiments used the same speech data set described in section 3.1.1 on page 33. We split the speech data set into a training set with 544 utterances and their labels and a test set of the remaining 575 utterances. The split preserved an even distribution of speakers in both sets, but was otherwise random.

3.4.2 Design of Case Combinations

The goal of the experiment was to explore the performance of different choices of feature sets, window sizes, frame periods, and sizes of the mixture-of-Gaussian models. As described in the previous section (page 41), we have nine combinations of feature sets to explore.

Since the average fundamental frequency of female speech is around 220 Hz, the period of the voiced sounds is approximately 4.5 ms. We therefore chose a set of 9 frame periods from 4 ms to 12 ms in steps of 1 ms. The smallest frame period was chosen to be only marginally smaller than the average period of the fundamental frequency. The largest frame period was chosen to be above the default frame period (10 ms) suggested by the HTK toolkit.

We chose a set of 9 possible window sizes from 10 ms to 30 ms in steps of 2.5 ms. With the constraint that the window size should be at least as long as the frame period, there are 79 ($= 9 \times 9 - 2$) possible combinations of window size and frame period and therefore 711 ($= 79 \times 9$) combinations of the speech encoding parameters.

We also explored 5 different sizes for the mixture-of-Gaussian models: 1, 2, 4, 8, and 16. There were therefore a total of 3,555 ($= 711 \times 5$) different possible combinations of parameters to evaluate.

We refer to the different sets of parameters by hyphenated codes such as "9-10-MFCC-E-D-A-4" where the first number is the frame period, the second number is the window size, the middle letters specify which of the MFCC features were used, and the final number is the size of the mixture-of-Gaussian models.

3.4.3 Training and Testing Process

For each combination of parameters, a set of phoneme level HMMs was trained on the utterances (and their labels) in the training set. During the training process, each utterance was encoded and the relevant features were extracted based on the choice of features, window size, and frame period. Each HMM state was modelled initially by a mixture-of-Gaussians of size 1 and trained using four cycles of the Baum-Welch training algorithm [85]. The maximum size of the mixture-of-Gaussians was then doubled and two cycles of the Baum-Welch re-estimation were applied. This was repeated until the maximum size of 16 was reached.

After obtaining the phoneme level HMMs, the testing process was conducted by applying these HMMs to the utterances in the test set using forced alignment and the Viterbi algorithm [85]. The testing process generated a set of auto-labelled phonemes (phoneme name, start, and end time) for each utterance. These auto-labelled phonemes were then be compared to the hand-labelled phonemes to measure the accuracy of the HMMs.

3.4.4 Experiment Configuration

Since training and testing a set of HMM models is a computationally intensive task, it would not have been feasible to run this exhaustive experiment on a single computer. Instead, we constructed a distributed server-client computing system to run the experiment. The system consisted of one Sun Fire 280R Server and 22 1.8 GHz Pentium 4 workstations with 128MB RAM running NetBSD, as shown in Figure 3.7. Even with these 22 computers, the experiment took more than 48 hours to complete.

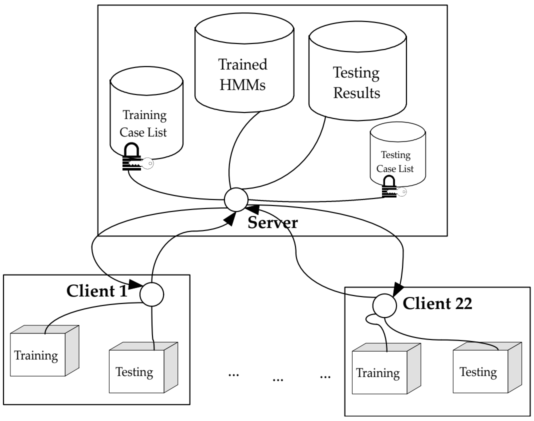


Figure 3.7: Experiment configuration.

There are two central synchronised lists on the server, one containing all training cases (one case for each combination of speech encoding parameters) and the other for testing cases. To reduce the network traffic, the utterances in the training and test data sets were pre-distributed to each client (workstation). The training time varies with different training cases. Therefore, instead of pre-assigning a fixed number of training cases to each client, we created connections between the server and the 22 clients so that the clients can train or test any case. Clients repeatedly request a training case (a combination of parameters) from the server, perform the training process, then send the trained HMMs back to the server, which are ready to be applied to the testing cases. Once the list of training cases is empty, each client starts requesting testing cases from the server, performing the testing and sending the resulting set of auto-labelled utterances back to the server. The effect of this distributed process is that no clients are idle until the very end of the experiment.

3.4.5 Performance Evaluation

To measure the recognition performance of a case, the system compares the auto-labelled phonemes in each utterance of the test set against the hand-labelled phonemes.

In the context of our speech analyser system, the most important requirement on the recognition system is the accuracy of the time boundaries of the auto-labelled phonemes. The simplest way of measuring this accuracy would be to measure the average error, where the error is the time difference between the boundary of the auto-labelled phoneme and the hand-labelled phoneme. However, we suspect that large errors will be much more significant for the rest of the speech analyser than small errors. Also, the nature of continuous speech is such that determining hand-labelled phoneme boundaries necessarily involves a subjective judgement. This means that small errors should not affect the accuracy measure. We therefore set a threshold and define the boundary accuracy to be the percentage of phonemes for which the difference between the auto-labelled boundary and the hand-labelled boundary is less than the threshold.

Obviously, the actual accuracy measure will vary with different thresholds — with a sufficiently high threshold, all the cases would have a 100% accuracy. However, the purpose of the accuracy measure is to compare the performance of different cases, so only the relative value of the accuracy measure is important, and the threshold value is not too critical. As is common among other studies [58, 64, 79], we chose 20 ms for the threshold of the performance evaluation. We also looked at the effect on the results of changing this threshold in either direction.

We were also interested in the sources of boundary time errors. We hypothesised that:

Hypothesis 3.4.1 *Some of the boundary time errors might be due to the recogniser misclassifying phonemes during the recognition process.*

We therefore performed a more standard recognition accuracy evaluation on the best performing HMM, measuring the fraction of phonemes in the test set that were misclassified by the HMM.

3.5 Results and Discussion

This section presents the relative recognition performance of the cases and gives some further analysis of the results.

3.5.1 Best Parameter Combinations

Using the threshold measure described in the previous section, we calculated the relative phoneme boundary accuracy of the HMM speech recogniser with 3,555 different combinations of parameters. Since the vowels play a more important role in the later stages of the speech analyser, we also calculated the relative accuracy results for vowels only (measuring just the end boundary of the vowel). The best 50 results and the worst result are given in Tables 3.2 and 3.3.

The best performance (87.07% accuracy for 10-12.5-MFCC-O-D-A-4) is considerably better than the worst performance (69.06% accuracy for 4-30-MFCC-8) in the same training and testing environment. So the choice of parameters, for the HMM design and the speech encoding process, is very important. Therefore, Hypothesis 2.1.1 (page 17) is supported by our results.

There are also several other observations that can be made from the results.

- There is a clear advantage in using the derivative (D) and acceleration (A) features. For all phonemes, all of the top 7.06% (251/3555) of the cases have the acceleration features, and all of the top 43.76% (1556/3555) of the cases have the derivative features. For vowels alone, all of the top 10.44% (371/3555) of the cases have the accelera-

Table 3.2: Best and worst parameter choices by boundary timing accuracy.

rank	Case	Boundary Accuracy	rank	Case	Boundary Accuracy
1	10-12.5-MFCC-O-D-A-4	87.07%	27	10-17.5-MFCC-O-D-A-16	86.41%
2	10-15-MFCC-O-D-A-8	87.00%	28	10-17.5-MFCC-E-D-A-4	86.40%
3	10-17.5-MFCC-O-D-A-4	86.99%	29	10-22.5-MFCC-O-D-A-2	86.38%
4	10-12.5-MFCC-O-D-A-8	86.99%	30	10-10-MFCC-E-D-A-16	86.38%
5	10-15-MFCC-O-D-A-4	86.97%	31	10-22.5-MFCC-O-D-A-4	86.37%
6	10-17.5-MFCC-O-D-A-8	86.87%	32	10-15-MFCC-E-D-A-16	86.37%
7	10-20-MFCC-O-D-A-4	86.84%	33	11-17.5-MFCC-O-D-A-4	86.37%
8	10-12.5-MFCC-O-D-A-2	86.83%	34	10-17.5-MFCC-O-D-A-1	86.34%
9	10-15-MFCC-O-D-A-2	86.79%	35	10-12.5-MFCC-E-D-A-8	86.31%
10	10-12.5-MFCC-O-D-A-1	86.70%	36	9-15-MFCC-O-D-A-4	86.30%
11	10-20-MFCC-O-D-A-8	86.69%	37	10-20-MFCC-E-D-A-8	86.30%
12	10-12.5-MFCC-O-D-A-16	86.63%	38	10-20-MFCC-O-D-A-1	86.29%
13	10-20-MFCC-O-D-A-2	86.61%	39	9-17.5-MFCC-O-D-A-4	86.29%
14	10-17.5-MFCC-O-D-A-2	86.60%	40	9-17.5-MFCC-O-D-A-8	86.26%
15	10-15-MFCC-E-D-A-8	86.56%	41	10-22.5-MFCC-E-D-A-8	86.26%
16	11-15-MFCC-O-D-A-4	86.51%	42	10-12.5-MFCC-E-D-A-16	86.23%
17	10-10-MFCC-O-D-A-8	86.50%	43	10-22.5-MFCC-E-D-A-4	86.23%
18	10-15-MFCC-E-D-A-4	86.49%	44	11-15-MFCC-O-D-A-8	86.22%
19	10-10-MFCC-O-D-A-4	86.49%	45	11-20-MFCC-O-D-A-4	86.20%
20	10-15-MFCC-O-D-A-1	86.47%	46	10-10-MFCC-O-D-A-2	86.19%
21	10-15-MFCC-O-D-A-16	86.44%	47	11-17.5-MFCC-O-D-A-8	86.17%
22	10-20-MFCC-E-D-A-4	86.44%	48	10-10-MFCC-E-D-A-8	86.17%
23	9-15-MFCC-O-D-A-8	86.44%	49	10-17.5-MFCC-E-D-A-16	86.17%
24	10-17.5-MFCC-E-D-A-8	86.43%	50	10-25-MFCC-O-D-A-4	86.14%
25	10-22.5-MFCC-O-D-A-8	86.43%	...	...	...
26	10-12.5-MFCC-E-D-A-4	86.42%	3,555	4-30-MFCC-8	69.06%

tion features, and all of the top 26.86% (955/3,555) of the cases have the derivative features.

- A frame period around 10 ms and a window size around 15 ms appear to give the best performance over all phonemes, but a larger frame period around 11 ms to 12 ms and a larger window size around 17.5 ms is better for performance on vowels alone.
- The 0th Cepstral (O) feature is clearly better than Energy (E) feature. There are only 15 of the top 50 cases for all phonemes and only one

Table 3.3: Best and worst parameter choices by boundary timing accuracy of vowels only.

rank	Case	Boundary Accuracy	rank	Case	Boundary Accuracy
1	11-15-MFCC-O-D-A-4	86.98%	27	10-12.5-MFCC-O-D-A-2	86.40%
2	12-15-MFCC-O-D-A-4	86.94%	28	10-15-MFCC-O-D-A-2	86.40%
3	12-17.5-MFCC-O-D-A-4	86.86%	29	11-20-MFCC-O-D-A-8	86.37%
4	11-17.5-MFCC-O-D-A-8	86.81%	30	11-17.5-MFCC-O-D-A-2	86.37%
5	12-12.5-MFCC-O-D-A-4	86.81%	31	11-15-MFCC-O-D-A-2	86.33%
6	10-17.5-MFCC-O-D-A-4	86.73%	32	12-20-MFCC-O-D-A-16	86.32%
7	12-20-MFCC-O-D-A-4	86.72%	33	12-15-MFCC-O-D-A-1	86.32%
8	11-15-MFCC-O-D-A-8	86.71%	34	12-12.5-MFCC-O-D-A-2	86.32%
9	11-17.5-MFCC-O-D-A-4	86.66%	35	11-12.5-MFCC-O-D-A-8	86.31%
10	11-12.5-MFCC-O-D-A-4	86.65%	36	12-20-MFCC-O-D-A-8	86.31%
11	12-20-MFCC-O-D-A-2	86.64%	37	12-25-MFCC-O-D-A-2	86.29%
12	12-12.5-MFCC-O-D-A-8	86.64%	38	12-15-MFCC-O-D-A-16	86.29%
13	12-22.5-MFCC-O-D-A-4	86.61%	39	12-25-MFCC-O-D-A-4	86.28%
14	12-17.5-MFCC-O-D-A-8	86.59%	40	12-17.5-MFCC-O-D-A-1	86.26%
15	12-17.5-MFCC-O-D-A-2	86.55%	41	11-12.5-MFCC-O-D-A-1	86.25%
16	12-15-MFCC-O-D-A-2	86.55%	42	10-12.5-MFCC-O-D-A-1	86.24%
17	10-17.5-MFCC-O-D-A-8	86.51%	43	12-17.5-MFCC-E-D-A-4	86.24%
18	12-17.5-MFCC-O-D-A-16	86.47%	44	12-20-MFCC-O-D-A-1	86.23%
19	10-15-MFCC-O-D-A-4	86.46%	45	10-12.5-MFCC-O-D-A-8	86.22%
20	12-15-MFCC-O-D-A-8	86.45%	46	12-12.5-MFCC-O-D-A-1	86.21%
21	10-12.5-MFCC-O-D-A-4	86.44%	47	11-15-MFCC-O-D-A-16	86.20%
22	12-22.5-MFCC-O-D-A-2	86.42%	48	11-17.5-MFCC-O-D-A-16	86.20%
23	11-20-MFCC-O-D-A-4	86.42%	49	11-22.5-MFCC-O-D-A-4	86.15%
24	11-12.5-MFCC-O-D-A-2	86.40%	50	10-20-MFCC-O-D-A-4	86.13%
25	11-20-MFCC-O-D-A-2	86.40%	...	...	...
26	10-15-MFCC-O-D-A-8	86.40%	3,555	4-30-MFCC-16	68.10%

of the top 50 cases for vowels using Energy.

- As shown in Table 3.4, there seems to be a preference towards a size of 4 for the mixture-of-Gaussians for vowels. Although using the size of 8 is much better for all phonemes than for vowels, the advantage of using the size of 4 still exists.
- The case (10-12.5-MFCC-O-D-A-4), which resulted in the best performance for all phonemes, is ranked 21st for vowels only. The case (11-

Table 3.4: Summary of the mixture-of-Gaussians for the top 50 cases.

Size	For All Phonemes	For Vowels Only
1	8% (4 out of 50)	12% (6 out of 50)
2	12% (6 out of 50)	24% (12 out of 50)
4	34% (17 out of 50)	32% (16 out of 50)
8	32% (16 out of 50)	22% (11 out of 50)
16	14% (7 out of 50)	10% (5 out of 50)

15-MFCC-O-D-A-4), which achieved the highest accuracy for vowels only, is ranked 16th for all phonemes. However, the difference in relative accuracy over the top 50 cases is less than 1%. So the exact choice of E vs O, frame period, window size and number of Gaussians, within the ranges above, does not appear to be very critical.

- Changing the threshold to 8 ms or 32 ms makes no difference to the strong advantage of the Derivative and Acceleration features. However, the preferred frame period and window size are slightly smaller for the 8 ms threshold, and slightly larger for the 32 ms threshold. Also for the 8 ms threshold, there is a preference for the Energy feature rather than the 0th cepstral feature for vowels.

The outcome of this experiment provides a clear recommendation for the parameters we should use for the speech analyser system: using 0th Cepstral, Derivative and Acceleration features, along with a frame period of 11 ms, a window size of 15 ms, and a mixture of 4 Gaussians should minimise the boundary timing errors on the vowels. If in later work the boundary timing differences of less than 20 ms are significant, we would then need to use the Energy feature, and a smaller frame period and window size.

3.5.2 Recognition Accuracy

The results above focused on the accuracy of the boundaries of the auto-labelled phonemes. The second evaluation attempted to identify some possible sources of the boundary errors by counting the kinds of phoneme recognition errors made by the recogniser using the best performing HMMs. There are three kinds of phoneme recognition errors:

- *substitution*, where the auto-labelled phoneme is different from the hand-labelled phoneme.
- *insertion*, where the auto-labelling includes an additional phoneme that is not present in the hand-labelling.
- *deletion*, where the auto-labelling does not contain a phoneme that is present in the hand-labelling.

Table 3.5 shows recognition errors of each category for the highest ranked HMM (10-12.5-MFCC-O-D-A-4) applied to the test set. Since insertion and deletion errors will almost certainly result in boundary time errors of phonemes adjacent to the error, in addition to the phoneme inserted or deleted, the nearly 6.45% of insertion or deletion phonemes is a non-trivial cause of the approximately 13% boundary timing error rate in Table 3.2. The substitution errors may or may not result in boundary timing errors. Therefore, Hypothesis 3.4.1 (page 45) is supported.

Table 3.5: Recognition errors for 10-12.5-MFCC-O-D-A-4.

Kind of Error	Fraction of phonemes
Insertion errors	5.33% (1372 out of 25723)
Deletion errors	1.12% (288 out of 25723)
Substitution errors	4.32% (1111 out of 25723)

The recognition system uses forced alignment recognition in which the system knows the target sentence and uses a dictionary of alternative word pronunciations to determine the expected phonemes. Insertion and deletion errors will generally occur when the actual pronunciation by the speaker does not match any of the pronunciations in the dictionary: the speaker drops a phoneme or includes an extra phoneme, and the system is forced to align the dictionary pronunciation with the actual pronunciation. These errors are due primarily to inadequacies in the dictionary, rather than to the HMM models. Substitution errors will result from pronunciations by the speaker that are not in the dictionary, but also may result from poor HMM phoneme models if the dictionary gives two alternative pronunciations, and the HMM for the wrong phoneme matches the speech signal better than the HMM for the correct phoneme.

3.6 Chapter Summary

In this chapter, we have described an HMM based speech recogniser using the forced alignment technique. The central requirement on the recogniser is that it can accurately identify the boundaries of the phonemes, especially vowels, in a speech signal.

We have reported on an experiment that exhaustively explored a space of parameter values for the recogniser. This included the parameters of the encoding of the speech signal and the size of the statistical models in the states of the phoneme HMMs. The results of the experiment provide clear recommendations for the choice of frame period, window size, MFCC features, and the statistical model in order to minimise the significant phoneme boundary errors. Hypotheses 2.1.1 (page 17) and 3.4.1 (page 45) have been supported.

The results of the experiment also expose the limitations of the dictionary used for forced alignment.

Chapter 4

Stress Detection Stage

This chapter discusses the stress detection stage as highlighted in Figure 4.1. Section 4.1 gives a more detailed overview of the stress detection stage, including the procedure for training a stress detector. Section 4.2 discusses the issues, including the calculation and normalisation methods of features, especially vowel quality features. Section 4.3 explores the performances of two different classification techniques — Decision Trees and Support Vector Machines. Section 4.4 presents the experimental results. Section 4.5 summarises this chapter. Note that part of this chapter is taken from [82].

4.1 Overview

The stress detection stage is the second stage of Span. We decided to detect rhythmic stress in users' speech instead of lexical stress because rhythmic stress is more important than lexical stress in good speech [77]. This stage performs rhythmic stress detection using a binary classifier. The rhythmic stress detection task is to classify vowel segments in the user's speech labelled at the phoneme level as *stressed* or *unstressed*. The classifier is trained from a speech data set, hand labelled at the phoneme level with the stress status marked.

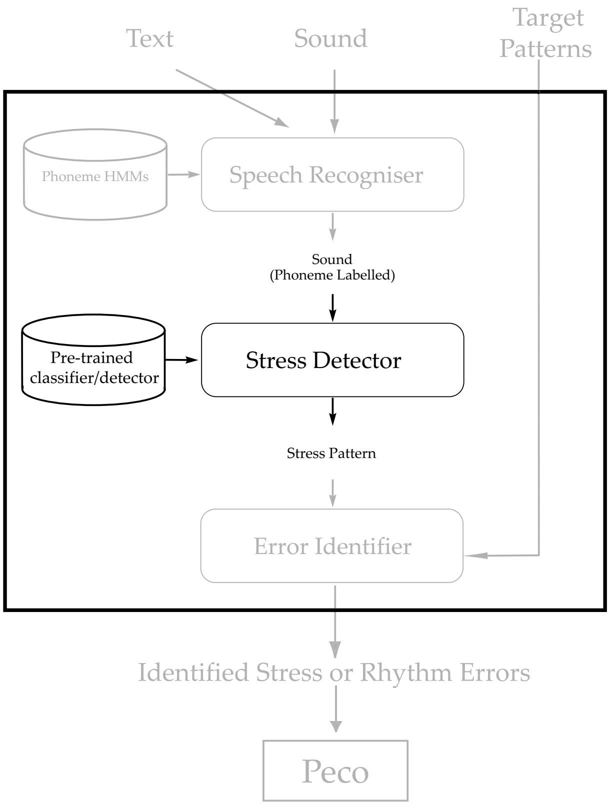


Figure 4.1: Overview of Span: detecting stress.

Note that the classification accuracy is critical to the final stage of Span, which identifies the stress or rhythm errors in the user's speech. If vowel segments are classified incorrectly at the second stage, then the generated user's stress pattern will be inaccurate. Consequently, the stress or rhythm errors reported by the final stage will not correctly indicate the user's prosodic problems.

4.1.1 The Training Element

The classifier training procedure is illustrated in Figure 4.2.

There are three related data sets as inputs to the training elements. Two of them are the same data sets used in the training element of the speech

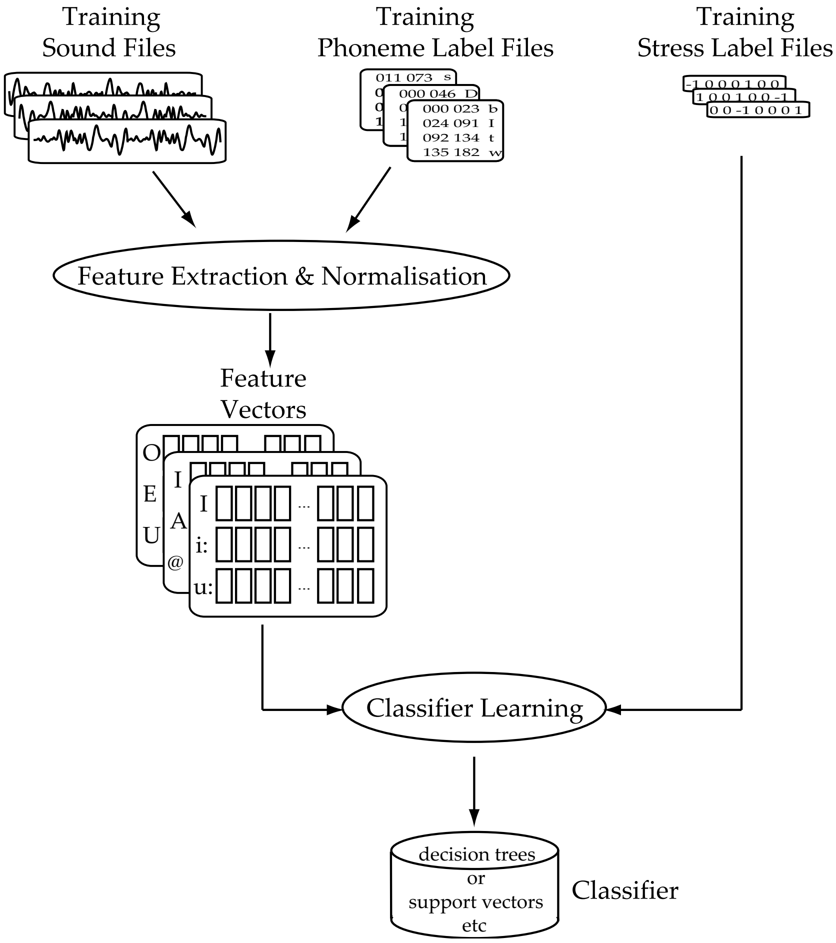


Figure 4.2: The training element.

recogniser. The third data set is a stress label data set.

The stress label data set contains stress label files corresponding to the phoneme files. Each sound file is now associated with a phoneme label file and a stress label file. We use -1, 1, and 0 to represent the stress status for each phoneme, where -1 means unstressed, 1 means stressed, and 0 means the phoneme is not a vowel and a stress mark is not applicable.

As shown in Figure 4.2, a feature extraction and normalisation process takes the training sound files and the phoneme label files, and produces sets of feature vectors for each vowel segment in each sound. A classifier learning procedure is then applied to construct a stress classifier, which can be represented by either decision trees or support vectors (or other representations used by other machine learning techniques), from the sets of feature vectors and the corresponding stress labels.

4.1.2 The Performance Element

The classification performance element is illustrated in Figure 4.3.

The phoneme labelled sound output from the speech recognition stage becomes the input of the performance element. The same feature extraction and normalisation process as used in the training element produces feature vectors representing vowel segments in the user's speech. By using the trained classifier, each feature vector is analysed and the corresponding vowel segment is classified as stressed or unstressed. Finally a stress pattern for the user's speech is constructed.

4.2 Feature Extraction and Normalisation

Although stress is generally seen as a property of the syllables in an utterance rather than of just the vowels, we hypothesise that:

Hypothesis 4.2.1 *The features we used in our study are largely carried by the vowel as the nucleus of the syllable. The features extracted from vowels can significantly contribute to rhythmic stress detection.*

Each vowel is analysed in several different ways to extract a set of features that can be passed to the stress classifier. Since duration, amplitude, pitch and vowel quality are the parameters that have been shown to cue the perception of stress differences in English [2, 18, 29, 46], the features we need to extract are related to these parameters.

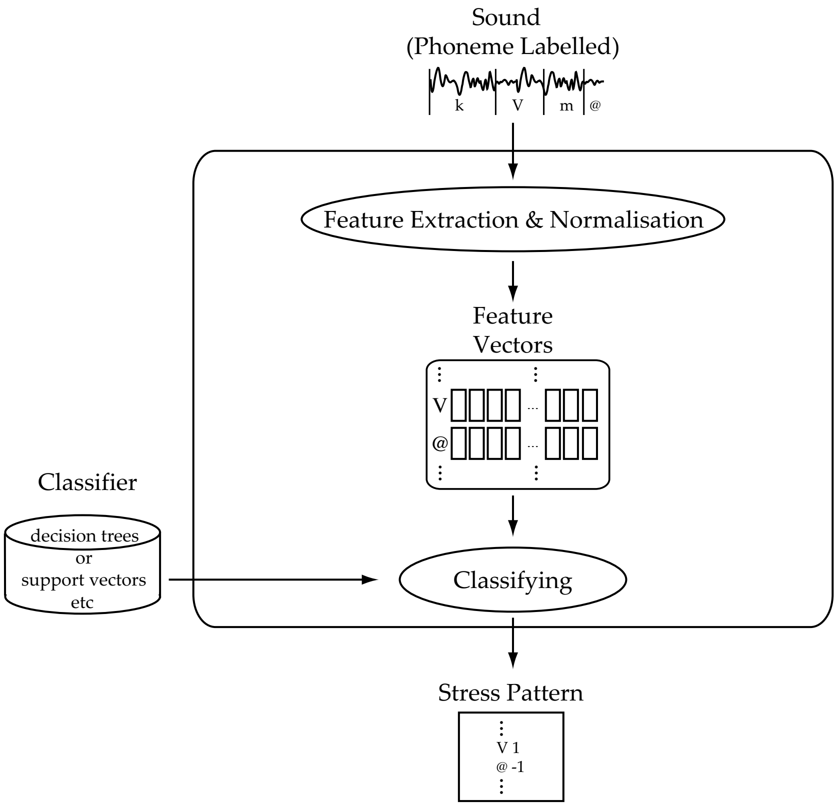


Figure 4.3: The performance element.

There are many alternative measurements of prosodic features that can be extracted, and also many ways of normalising these features in order to reduce variation due to differences between speakers, recording situations or utterance contexts. However, there is no standard way to extract and normalise vowel quality features.

4.2.1 Duration Features

Vowel durations can be directly measured from the output of the forced alignment recogniser since the recogniser identifies the start and end points of the vowels. The measurements are not completely reliable since it is hard for the recogniser to precisely determine the transition point between two phonemes that flow smoothly into each other. Furthermore, some short vowels may be inaccurately reported if they are shorter than the minimum number of frames specified for a phoneme in the system.

The absolute value of the duration of a vowel segment is influenced by many factors other than stress, such as the intrinsic durational properties of the vowel, the speech rate of the speaker, and local fluctuations in speech rate within the utterance. Therefore the absolute duration of the vowel segment is not a useful feature. We require a normalised duration that measures the length of the vowel segment relative to what would "normally" be spoken by an "average" speaker. To reduce the impact of these contextual properties, we applied three different levels of normalisation to the raw duration values.

The first level normalisation reduces the effect of speech rate variation between speakers. To normalise, we need to compare the length of an utterance to the "expected" length of that utterance. To compute the latter, we first use the training speech data set to calculate the average duration of each of the 20 vowel phonemes of NZ English. We then compute the expected utterance length by summing the average durations of the vowel types in the utterance, and the actual utterance length by summing the actual durations of the vowel segments in the utterance. We can then normalise the durations of each vowel segment by multiplying by the expected utterance length divided by the actual utterance length.

The second level normalisation removes effects of variation in the durations of the different vowel phonemes. Each phoneme has an intrinsic duration — long vowels and diphthongs normally have longer durations than short vowels. There are several possible ways to normalise for intrinsic vowel duration. One method is to normalise the vowel segment duration by the average duration for that vowel phoneme, as measured in the training data set. Another method is to cluster the 20 vowel phonemes into three categories (short vowel, long vowel and diphthong) and normalise vowel segment durations by the average duration of all vowels in the relevant category. We consider both methods.

The third level normalisation removes the effect of the variation in speech rate at different parts of a single utterance. To remove this influence, the result of the second level normalisation is normalised by a weighted average duration of the immediately surrounding vowel segments.

Based on the three levels of normalisation, we computed five duration features for each vowel segment:

- *Utterance normalised duration*: the absolute duration normalised by the length of the utterance;
- *Phoneme normalised duration*: the duration normalised by the length of the utterance and the average duration of the vowel type;
- *Category normalised duration*: the duration normalised by the length of the utterance and the average duration of the vowel category;
- *Phoneme neighbourhood normalised duration*: the vowel type normalised duration further normalised by the durations of neighbouring vowels;
- *Category neighbourhood normalised duration*: the vowel category normalised duration further normalised by the durations of neighbouring vowels.

4.2.2 Amplitude Features

The amplitude of a vowel segment can be measured from the speech signal, but since amplitude changes during the vowel, there are a number of possible measurements that could be made — maximum amplitude, initial amplitude, change in amplitude, etc. A measure commonly understood to be a close correlate to the perception of amplitude differences between vowels is the root mean square (RMS) of the amplitude values across the entire vowel. This is the measure chosen as the basis of our amplitude features. As with the duration features, amplitude is influenced by a variety of factors other than stress, including speaker differences and differences in recording conditions as well as changes in amplitude across the utterance. We therefore need to normalise measured amplitude to reduce variability introduced by these effects. We apply two levels of normalisation to obtain two amplitude features.

Our first level normalisation of amplitude takes into account global influences such as speaker differences and the recording situation, by normalising the RMS amplitude of each vowel segment against the overall RMS amplitude of the entire utterance.

Our second level normalisation considers local effects at different parts of the utterance and normalises the vowel amplitude against a weighted average amplitude of the immediately surrounding vowel segments.

4.2.3 Pitch Features

Like amplitude, pitch can vary over the course of the vowel segment and is influenced by a variety of different factors, including the basic pitch of the speaker's voice. To reduce the effects of speaker differences, we normalise the pitch measurement of a vowel segment by the average pitch of the entire utterance. A pitch calculation algorithm introduced in [9] was used in our system.

The change in pitch over the vowel segment is at least as important as the pitch level of the vowel, but it is not clear exactly which properties of pitch are most significant for determining stress. Therefore, we extracted 10 different pitch features, including not only the average normalised mean pitch value of a vowel segment, but other features intended to capture changes in pitch. The 10 pitch features of a vowel segment are summarised as follows:

- *Normalised mean pitch*: the mean pitch value of the vowel normalised by the mean pitch of the entire utterance.

- *Normalised pitch value at the start point*: the pitch value at the start point of the vowel divided by the mean pitch of the utterance.
- *Normalised pitch value at the end point*: the pitch value at the end point of the vowel divided by the mean pitch of the utterance.
- *Normalised maximum pitch value*: the maximum pitch value of the vowel divided by the mean pitch of the utterance.
- *Normalised minimum pitch value*: the minimum pitch value of the vowel divided by the mean pitch of the utterance.
- *Relative pitch difference*: the difference between the normalised maximum and minimum pitch values. A negative value indicates a falling pitch and a positive value indicates a rising pitch.
- *Absolute difference*: the magnitude of the *Relative difference*, which is always positive.
- *Pitch trend*: the sign of the *Relative difference* — +1 if the pitch "rises" over the vowel segment, -1 if it "falls", and 0 if it is "flat".
- *Boundary Problem*: a boolean attribute — true if the pitch value at either the start point or the end point of the vowel segment cannot be detected.
- *Length Problem*: a boolean attribute — true if the vowel segment is too short to compute meaningful minimum, maximum, or difference values.

4.2.4 Vowel Quality Features

As mentioned earlier, vowel quality features are more difficult to extract since there is no standard approach that can be used. We developed a novel method that uses the vowel HMM models, which were obtained in the first stage of Span, to re-recognise the vowel segments of the utterance and extract vowel quality features.

Centralised vowels (see page 22) are associated with unstressed syllables, particularly the /ə/ vowel which is the primary reduced vowel. In NZ English, /ɪ/ is also pronounced very centrally and often acts as a reduced vowel. Among 20 English vowel types, excepting /ə/ and /ɪ/, there are 18 full vowels. Full vowels tend to be more peripheral, but can be associated with both stressed and unstressed syllables.

To determine whether a vowel segment represents a reduced vowel, we need to recognise the intended vowel phoneme, and also to determine whether it is pronounced more centrally than the norm for that vowel. Since our speech recogniser uses forced alignment, it only identifies the segments of the utterance that match each expected vowel best and does not identify how the speaker pronounced the vowel. For the prosodic features above, this is all that is needed, and using a full recogniser on the entire sentence would reduce the accuracy of the recognition. However, for measuring vowel quality, we need to know what vowel the speaker actually said, and how they pronounced it.

To determine the actual vowel quality of the vowels, we apply a very constrained form of full recognition to each of the vowel segments previously identified by forced alignment, and use the probability scores of the individual HMM phoneme models to compute several features that indicate whether the vowel is reduced or not. The algorithm is illustrated in Figure 4.4 and outlined below.

Step 1 Extract vowel segments from the utterance using forced alignment. Label each segment with the expected vowel phoneme label, based on the target sentence and the pronunciation dictionary.

Step 2 Encode each vowel segment into a sequence of acoustic parameter vectors, using a 15 ms Hamming window with a step size (frame period) of 11 ms. These parameters consist of 12 MFCC features and the 0′th cepstral coefficient with their first and second order derivatives. The values of these parameters were suggested from the experiment described in section 3.5.1 (page 46).

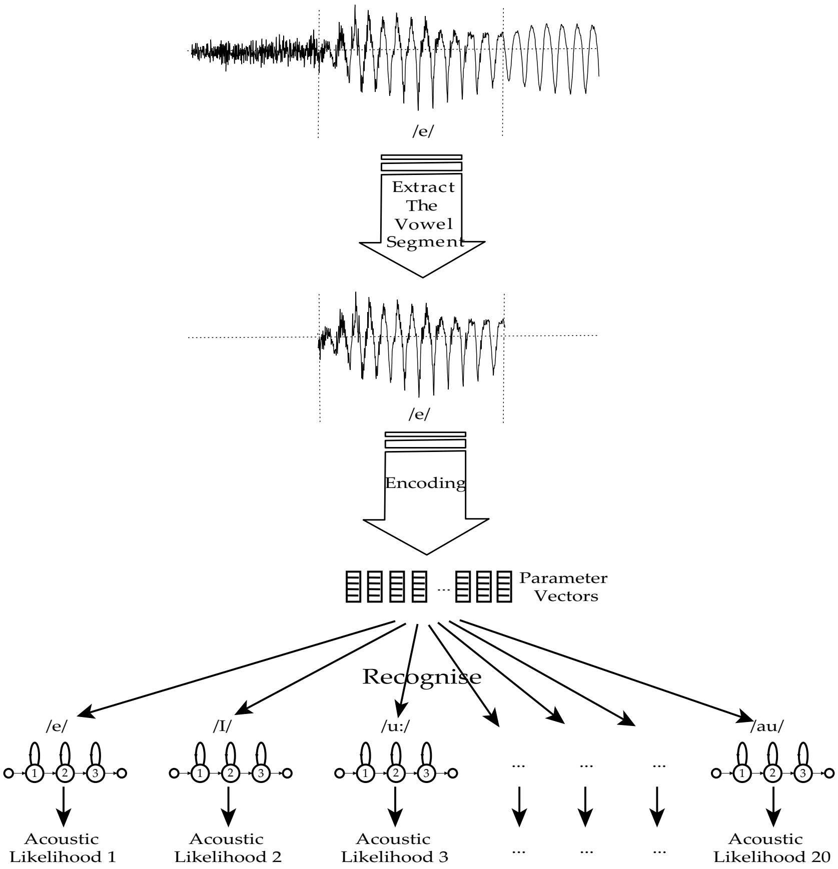


Figure 4.4: Vowel quality features processing.

Step 3 Feed the parameter vector sequence into the 20 pre-trained HMM vowel recognisers to obtain 20 normalised acoustic likelihood scores. Each score is the geometric mean of the acoustic likelihoods of all frames in the segment, as computed by the HMM recogniser. The scores are likelihoods that reflect how well the segment matches the vowel type of the HMM.

Step 4 Find the score of the expected vowel type S_e, the maximum score of any full vowel phoneme S_f and the maximum score of any reduced vowel phoneme S_r from the above 20 scores.

Step 5 Compare the scores of the best matching full vowel and the best matching reduced vowel to the score of the expected vowel in order to determine whether the expected vowel is pronounced as a full vowel or a reduced vowel. We compute four features, two of which measure the difference between the likelihoods, and two measure the ratio of the likelihoods. In each case, we take logarithms to reduce the spread of values.

$$R_d = \begin{cases} -\log(S_r - S_e) & \text{if } S_e < S_r \\ 0 & \text{if } S_e = S_r \\ \log(S_e - S_r) & \text{if } S_e > S_r \end{cases} \tag{4.1}$$

$$F_d = \begin{cases} -\log(S_f - S_e) & \text{if } S_e < S_f \\ 0 & \text{if } S_e = S_f \\ \log(S_e - S_f) & \text{if } S_e > S_f \end{cases} \tag{4.2}$$

$$R_r = \log(S_e/S_r) = \log S_e - \log S_r \tag{4.3}$$

$$F_r = \log(S_e/S_f) = \log S_e - \log S_f \tag{4.4}$$

Both difference and ratio measures have advantages and disadvantages. We explore which of the two approaches is better for the detection of rhythmic stress.

Step 6 Compute a boolean vowel quality feature, T, to deal with cases where the vowel segment is so short that F or R cannot be calculated. If the vowel segment is less than 33 ms, which is the minimum segment duration requirement of our HMM system, then the value of this attribute will be 1. Otherwise, -1. If this value is 1, we set F and R to 0.

4.3 Classifier Learning and Feature Evaluation

The goals of our experiments described below are to investigate whether it is feasible to build an effective automated stress detector for English utterances and to evaluate the different sets of features that we have extracted.

Our approach is to use two examples of standard machine learning tools — a decision tree constructor (C4.5) [62] and a support vector machine (LIBSVM) [13] – to construct stress detectors using these features, and to measure the performance of the resulting stress detectors. One reason for considering a decision tree constructor is that they can generate explicit rules that might help us identify which features were most significant for the stress detector. One reason for choosing a support vector machine constructor is that the support vector machines technology is relatively new. Support vector machines were originally designed for solving binary classification problems [16], which is just what the stress detector will perform. For margin classifiers (boosted DTs), which are also now considered to be the state of the art, we would explore it in the future.

4.3.1 Data Set

The experiments used a subset of the speech data that was described in the training element on page 53. The subset contains 60 utterances of ten distinct English sentences produced by six adult female NZ speakers, as part of the data set used in the first stage of Span (see page 33). The utterances were hand labelled at the phoneme level, and each vowel was labelled as *stressed* or *unstressed*. There are 703 vowels in the utterances. 340 are stressed and 363 unstressed. The prosodic and vowel quality features were extracted for each of these vowels.

4.3.2 Performance Evaluation

The task of our stress-detector is to classify vowels as stressed or unstressed. Neither stress category is weighted over the other, and so we use classification accuracy to measure the performance of each classifier.

Since the data set is relatively small, we applied the 10-fold cross validation method for training and testing the stress detectors. In addition, we repeat this training and testing process ten times. Our results below are the average results over the ten repetitions.

4.3.3 Experiment Design

As discussed earlier, we computed several sets of features and selected two learning algorithms for the construction of stress detectors. We designed three experiments to investigate a sequence of research questions.

To explore which subset of prosodic features is most useful for learning stress detectors for our data, the first experiment uses the two learning algorithms in conjunction with all seven different combinations of the prosodic features (D, A, P, D+A, D+P, A+P, and D+A+P, where D, A, and P are the sets of duration, amplitude, and pitch features, respectively).

To assess the contribution of vowel quality features to stress detection, the second experiment uses the two learning algorithms in conjunction with seven different combinations of the vowel quality features ($F_d + T$, R_d+T, $R_d + F_d + T$, $F_r + T$, R_r+ T, $R_r + F_r + T$, and $R_d + F_d + R_r + F_r + T$).

The third experiment investigates whether combining the prosodic features and the vowel quality features improves performance.

In these experiments, we also investigate whether scaling the feature values to the range [-1 . . . 1] improves performance.

For LIBSVM, we used a radial basis function kernel and a C parameter of 1.0. More information about kernels and parameters used in LIBSVM can be found in [13].

4.4 Results and Discussion

4.4.1 Experiment 1: Prosodic Features

The results for the prosodic features in the first experiment are shown in Table 4.1. Overall, the best results obtained by the LIBSVM are almost always better than those obtained by C4.5 for all feature combinations.

Table 4.1: Results for prosodic features.

Features	C4.5		LIBSVM	
	Unscaled	Scaled	Unscaled	Scaled
D	80.66	80.22	81.00	82.55
A	68.18	68.26	70.18	69.08
P	55.12	56.00	57.82	58.45
$D + A$	81.34	81.06	83.88	84.72
$D + P$	80.84	80.10	79.27	81.55
$A + P$	66.96	66.36	70.00	70.28
$D + A + P$	80.40	80.58	79.72	83.23

Scaled data led to better performance than unscaled data for LIBSVM in most cases, but this is not true for C4.5. For both LIBSVM and C4.5 , the combination of duration and amplitude features ($D + A$) produced the best results, which are 84.72% and 81.34% respectively. Adding the pitch features to this subset did not improve performance in any case. These results suggest that the set of features (D+A) is the best combination for our data set. While both decision trees and support vector machines are supposed to be able to deal with the redundant features, neither of them performed well at ignoring the less useful features in this experiment.

4.4.2 Experiment 2: Vowel Quality Features

The second experiment investigated the performance of the vowel quality features. The results are shown in Table 4.2. The vowel quality features alone achieved results that were very comparable to the performance of the prosodic features. The best result was 82.50%, which was achieved by LIBSVM using vowel quality features ($R_r + T$). This result was only 2.22% lower than the best result achieved by LIBSVM using prosodic features (D+A). Furthermore, the performance achieved by C4.5 using ($R_r + T$) (82.15%) is better than using ($D + A$) (81.34%). In addition, the following points can be noted.

Table 4.2: Results for vowel quality features.

Features	C4.5		LIBSVM	
	Unscaled	Scaled	Unscaled	Scaled
$F_d + T$	65.50	66.17	66.57	68.27
$R_d + T$	80.74	80.87	81.36	81.51
$F_d + R_d + T$	79.88	79.73	79.12	81.51
$F_r + T$	67.80	68.38	62.56	63.44
$R_r + T$	82.14	82.15	82.50	78.37
$F_r + R_r + T$	80.64	80.48	81.29	78.37
R_d + F_d + R_r + F_r + T	79.68	78.90	79.05	81.51

- The reduced vowel quality features R_d and R_r are more reliable than full vowel quality features F_d and F_r, which supports Hypothesis 2.2.1 (page 22).
- For LIBSVM, scaling is recommended when using likelihood differences for vowel quality features, ; if likelihood ratios are used, scaling is not needed.
- For LIBSVM, with a smaller number of scaled features — five features in total, the ability of handling redundant features is demonstrated in this experiment. The performance retains 81.51% whenever scaled features $R_d + T$ are included in the feature set.
- For C4.5, using the likelihood ratios is better than using the likelihood differences.
- For C4.5, in most cases, scaling produced slightly better results than non-scaling, regardless of whether differences or ratios were used, but the difference in performance between scaling and non-scaling was always very small.

4.4.3 Experiment 3: All Features

Table 4.3: Results for prosodic and vowel quality features.

Features	C4.5		LIBSVM	
	Unscaled	Scaled	Unscaled	Scaled
$C + V_d$	80.26	81.38	81.04	82.23
$C + V_r$	80.30	80.42	81.14	82.40
$C + V_d + V_r$	79.97	80.06	81.28	82.01

The third experiment was performed using the combination of all the prosodic features (C) and the vowel quality features using either the difference ($V_d = F_d + R_d + T$) or the ratio measure of vowel quality ($V_r = F_r + R_r + T$). As can be seen from Table 4.3, combining all features from the two sets did not improve the best performance on our data set over using either prosodic or vowel quality features alone. However, the result did demonstrate that the LIBSVM achieved better performance than the C4.5 on the data set, suggesting that SVM is more suitable for a relatively large data set with all numeric data.

For all the three experiments, C4.5 produced rules that were far more complex and much harder to interpret than expected. Given that most of the features are numeric, this was not too surprising.

4.5 Chapter Summary

In this chapter, we have explored rhythmic stress detection methods using SVM and DT techniques. We have also studied a range of prosodic features and vowel quality features, including feature extraction, normalisation and scaling. We have developed a novel method for measuring vowel quality features.

The results of the experiments suggest that we should use SVM to build the automatic rhythmic stress detector and duration and amplitude features to detect rhythmic stress in our data set.

The results of the experiments also indicated that SVM is more suitable for a relatively large data set with all numeric data compared to DT (C4.5).

Hypotheses 4.2.1 (page 55) and 2.2.1 (page 22) have been supported.

Chapter 5

Error Identification Stage

This chapter discusses the stress and rhythm error identification stage as highlighted in Figure 5.1. Section 5.1 gives a brief overview of the stress and rhythm error identification stage. Section 5.2 discusses the vowel sequence alignment problem. Section 5.3 discusses error identification in the user's stress pattern. Section 5.4 explores error identification in the user's rhythm pattern. Section 5.5 presents visualisation tools for users. Section 5.6 summarises this chapter.

5.1 Overview

The stress and rhythm error identification stage is the final stage of Span. As shown in Figure 5.2, it first aligns the vowel sequences in the target stress pattern and the user's stress pattern. It then performs stress error identification by comparing the stress status of a vowel in the user's speech with the corresponding status in the target. If no error is found, it then derives the user's rhythm pattern and performs a rhythm error identification by comparing the user's rhythm pattern with the target rhythm pattern. The reported errors from the error identifier are then fed into Peco, which uses the given information to provide useful personalised feedback to the user. Therefore, the reported errors must include the most

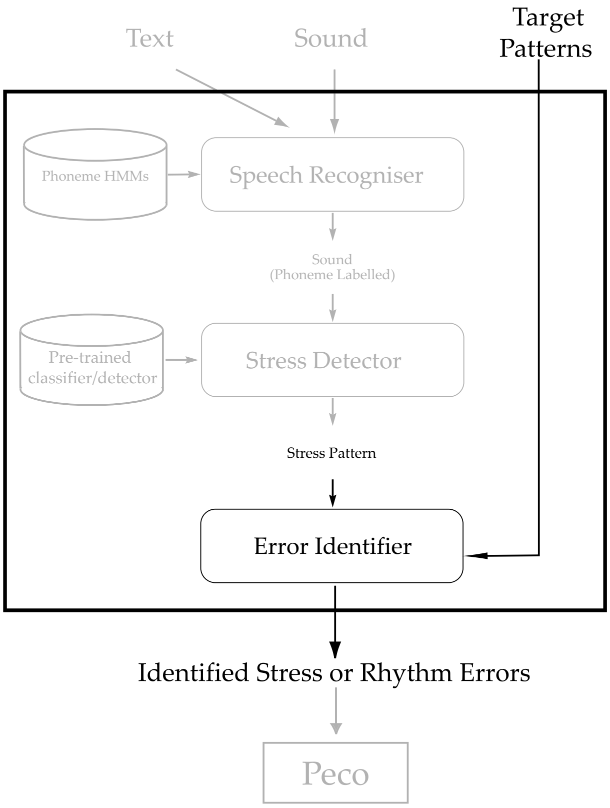


Figure 5.1: Overview of Span: identifying stress and rhythm errors.

critical problems in the user's speech but must not include redundant, consequential, or minor errors, which may prevent Peco from providing useful feedback.

5.2 Vowel Sequence Alignment

As described in the previous chapter, a stress pattern in an utterance consists of a sequence of vowels and their stress status. To identify whether the user stressed or unstressed vowels correctly, we need to check whether the status of each vowel in a user's speech matches the status of the corresponding vowel in the target speech. In order to do this, we first need to

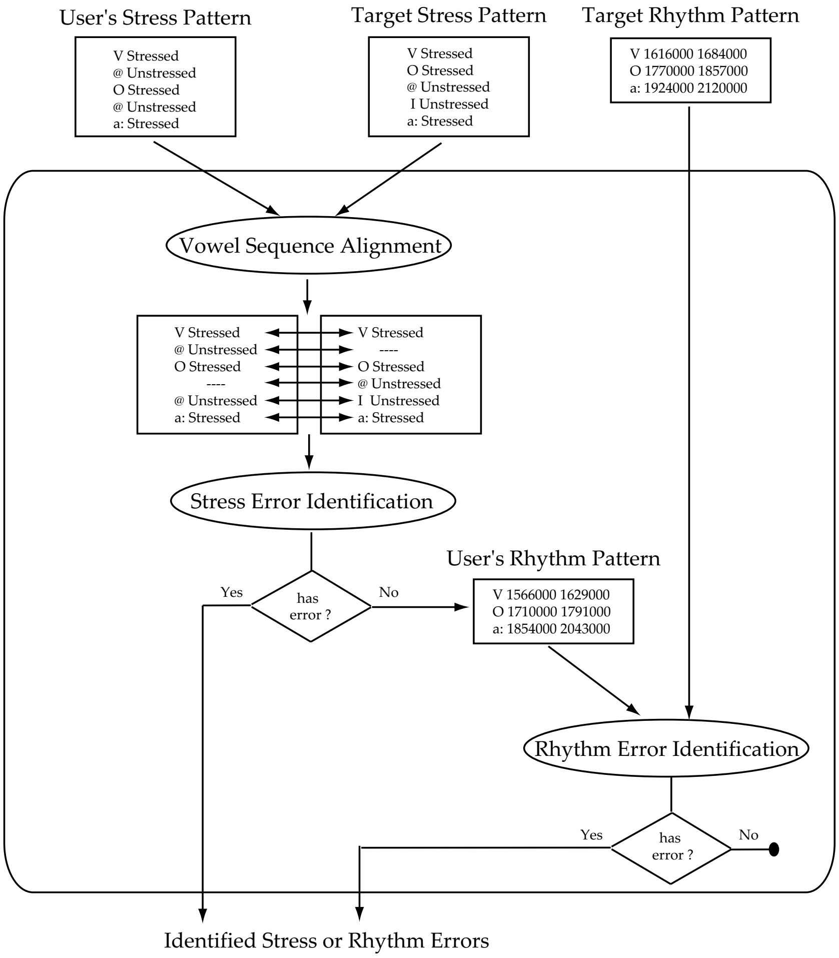


Figure 5.2: Overview of the error identifier.

properly align the two sequences of vowels.

As our speech recogniser uses the forced alignment technique, the best matched phoneme sequence aligned to the user's speech is from the word pronunciation dictionary regardless of what the user actually utters. Therefore if the dictionary does not include any alternatives (either acceptable pronunciations or mistakes), then the speech recogniser will produce the same phoneme sequence as the target one despite the errors the user may make. In this case the vowel sequence alignment will be trivial but the stress pattern in the user's speech may not be correctly identified since pronunciation errors such as insertion, deletion and substitution, may not be noticed and explored by the speech recogniser. If the dictionary does include alternatives, the speech recogniser may produce a phoneme sequence different from the target one. In this case, the vowel sequence alignment will be non-trivial and it could be very difficult to achieve the ideal alignment result.

5.2.1 Fundamentals of the Alignment Algorithm

We adapted a dynamic programming algorithm called the Needleman/Wunsch algorithm [55] to perform the vowel sequence alignment. The algorithm has three steps: *initialisation*, *scoring*, and *traceback*.

Assume that: (1) there are two sequences A and B, which have M and N elements, respectively; (2) the sequence A is used as the target.

Initialisation Create a matrix X with $M+1$ columns and $N+1$ rows so there are $(M+1)\times(N+1)$ cells in total. Each cell has a value and three backpointers. For each cell in the first row and first column of the matrix, values are initially filled with 0. All backpointers in all cells initially point to NULL.

For example, we have two sequences $A = \{V, @, O, @, \text{a:}, @\}$ and $B = \{V, O, @, I, \text{a:}\}$. The size of A is six and the size of B is five. We then create a matrix X with 6×7 cells as shown in Figure 5.3. Note that backpointers pointing to NULL are not shown in this figure (nor are they shown in Figures 5.4 and 5.5).

Scoring Fill the value in a cell (i, j) with $X_{i,j}$, the maximum global alignment score ending at the cell (i, j), where $1 \leq i \leq N$ and $i \leq j \leq M$.

$$X_{i,j} = Max[X_{i-1,j-1} + S_{i,j}, X_{i,j-1}, X_{i-1,j}] \tag{5.1}$$

Sequence A

Sequence B

		V	@	O	@	a:	@
	0	0	0	0	0	0	0
V	0						
O	0						
@	0						
I	0						
a:	0						

Matrix X

Figure 5.3: Alignment algorithm: initialisation.

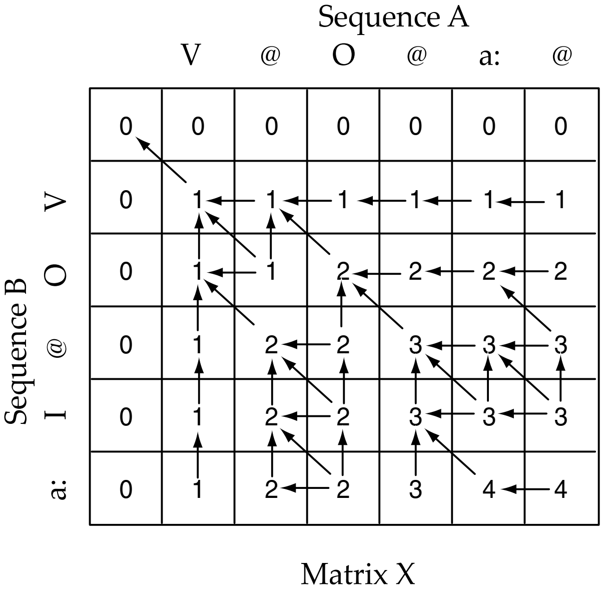
$S_{i,j}$ is defined as the local alignment score. $S_{i,j}$ is set to 1 if the vowel at position i of sequence B is the same as the vowel at position j of sequence A, otherwise it is 0.

If $X_{i,j} = X_{i-1,j-1} + S_{i,j}$, then set a backpointer to the cell $(i-1, j-1)$ in the cell (i, j). If $X_{i,j} = X_{i,j-1}$, then set a backpointer to the cell $(i, j-1)$ in the cell (i, j). If $X_{i,j} = X_{i-1,j}$, then set a backpointer to the cell $(i-1, j)$ in the cell (i, j). It is possible for each cell to have more than one non-null backpointer by the end of the scoring. That means there exist more than one possible path leading to the cell (i, j).

For example, Figure 5.4 illustrates the above matrix X after completing the scoring.

Traceback Following the backpointers in each cell, starting from the cell at the bottom right corner of the matrix X, move back to find a path and generate alignment results.

From a cell (i, j), if there exists a backpointer pointing to a cell $(i, j-1)$,



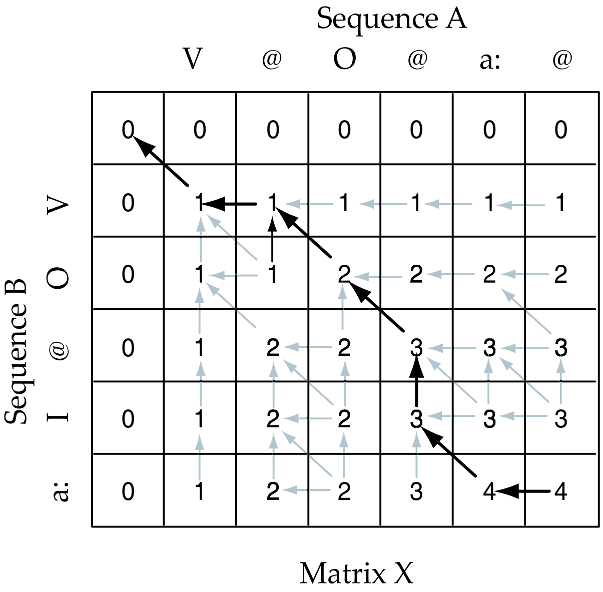
		V	@	O	@	a:	@
	0	0	0	0	0	0	0
V	0	1	1	1	1	1	1
O	0	1	1	2	2	2	2
@	0	1	2	2	3	3	3
I	0	1	2	2	3	3	3
a:	0	1	2	2	3	4	4

Figure 5.4: Alignment algorithm: scoring.

it means that the j^{th} element in the sequence A is an extra element or the sequence B has a deletion error. A "–" is inserted into the sequence B to represent the missing element. If there exists a backpointer pointing to a cell $(i-1, j)$, it means that the i^{th} element in the sequence B is an insertion error. A "–" is then inserted into the sequence A to expand its size. If there exists a backpointer pointing to a cell $(i-1, j-1)$, it means that the elements in sequence A and B are matched or are substitutions.

If from a cell (i, j), there are more than one backpointers pointing back to its predecessors that include the cell $(i-1, j-1)$, then the backpointers pointing to the cell $(i-1, j-1)$ will be chosen to form the traceback path. Otherwise one of the backpointers is arbitrarily chosen.

For the example used above, the traceback path and the alignment result are shown in Figure 5.5.



		V	@	O	@	a:	@
	0	0	0	0	0	0	0
V	0	1	1	1	1	1	1
O	0	1	1	2	2	2	2
@	0	1	2	2	3	3	3
I	0	1	2	2	3	3	3
a:	0	1	2	2	3	4	4

```
Sequence A:  V  @  O  @  --  a:  @
             |  |  |  |  |   |   |
Sequence B:  V  -- O  @  I   a:  --
```

Figure 5.5: Alignment algorithm: traceback.

5.2.2 Additional Issue

Since one of the tasks in this research is to automatically identify the stress pattern in the user's speech, the speech recogniser should identify most of the pronunciation differences in users' speech, especially the insertion and deletion errors, because these two types of errors directly affect the stress pattern and the rhythm pattern by changing the number of vowels. In order to identify most of the errors in an ESL student's speech, particularly for Mandarin speakers, we conducted a small experiment with the following changes made in the speech recogniser:

- Amend the word pronunciation dictionary by adding extra schwas at the end of some words or in some consonant clusters;

- Change the phoneme network construction by setting every phoneme as optional instead of standard compulsory mode.

After applying these changes, the results of the small experiment show that the recognised phoneme sequence reflects the actual pronunciation more precisely, which means more inserted or deleted phonemes can be identified in the user's incorrect speech. However, it makes the vowel sequence alignments much more difficult since the discovered insertions, deletions, and substitutions of phonemes cause more ambiguities.

For example, for a sentence containing a word "vitamins", with the target vowel sequence of "...ɪ ə ɪ...", and the user's vowel sequence of "...ə...", obviously there are errors in the user's speech. But by just examining the two vowel sequences, we cannot determine which is the true cause of errors from the following four possibilities:

- The /ə/ in the user's speech is a substitution of the first /ɪ/ in the target speech and the rest vowels are missing;
- The /ə/ in the user's speech matches with the middle /ə/ but the two /ɪ/ vowels are missing;
- The /ə/ in the user's speech is a substitution of the second /ɪ/ in the target speech and the previous two vowels are missing;
- The /ə/ in the user's speech is an insertion problem at the end of the previous word in the sentence, which commonly happens to a Mandarin speaker, and the target word "vitamins" is not pronounced at all.

One approach for reducing ambiguity and difficulty of the alignment is to align the two phoneme sequences of the user's speech and the target speech instead of just the two vowel sequences because the additional consonant phonemes can help to reduce the ambiguity and the difficulty.

However, our preliminary experimental results show that our standard algorithm can still find more than one possible alignment result. It cannot handle this issue properly.

5.2.3 Two-layer Alignment Algorithm

To address this additional problem, we perform the sequence alignment at two layers — the word level as well as the phoneme level. In order to do so, we first need to know which word each chunk of phonemes belongs to. There are at least two ways to obtain the word – phoneme mapping information. One way is to modify the speech recognition process so that the output contains both phoneme and word annotations. The other way is to perform an additional word level forced alignment and then map the words to phonemes by checking their time stamps. We chose to use the second method because the changes to the output of the first speech recognition stage would cause many significant changes in other later stages.

After obtaining the word – phoneme mapping, as shown in Figure 5.6, the recognised word sequence of the user's speech is aligned with the target word sequence. This is the first layer sequence alignment. We then perform the second layer alignment that aligns the phoneme sequences of each aligned word pair. After that, we filter out the non-vowel phonemes and obtain the aligned vowel sequences. This two-layer sequence alignment approach greatly reduces the ambiguity of the phoneme alignment process.

5.3 Stress Error Identification

After properly obtaining the two aligned vowel sequences, we are able to identify the stress errors in the user's speech by comparing the stress statuses of each aligned vowel pair.

When comparing the stress status of a vowel in the user's speech with

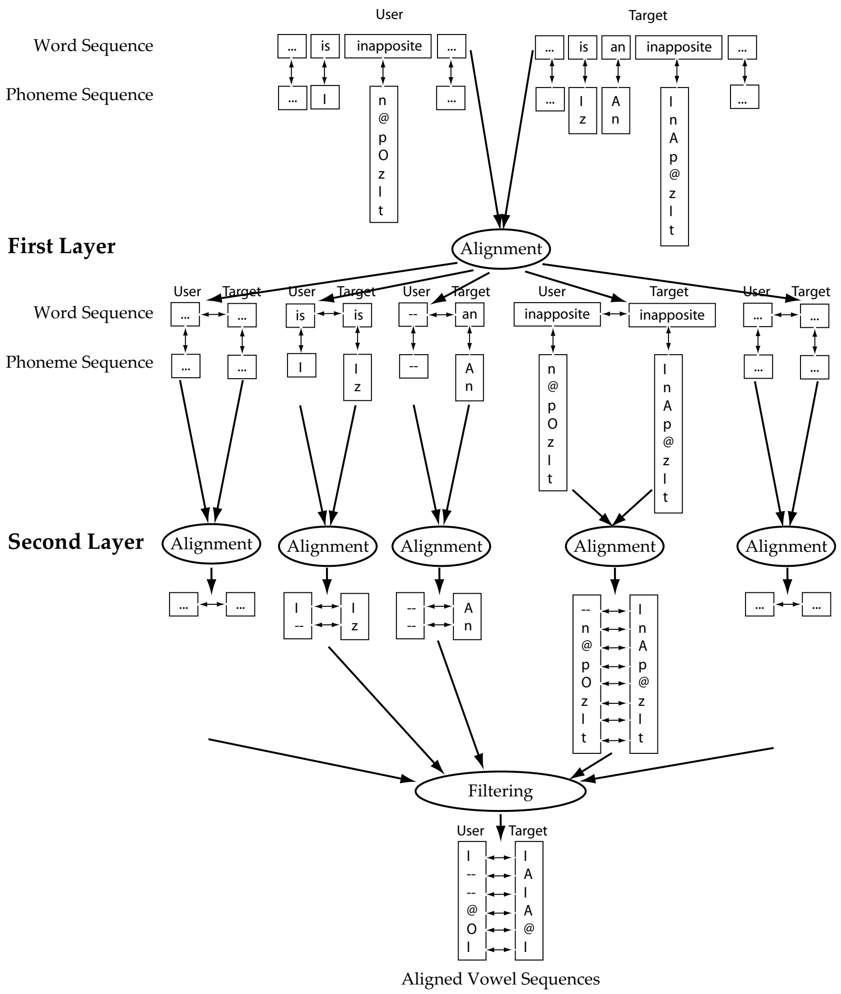


Figure 5.6: Two-layer vowel sequence alignment.

the corresponding vowel in the target speech, we can obtain the following kinds of situations:

1. Stressed – Stressed and Unstressed – Unstressed

   These are the ideal results. The stress status of a vowel in the user's speech matches the stress status of the aligned vowel in the target speech.

2. Stressed – Unstressed and Unstressed – Stressed

These are the worst cases. The stress statuses are just opposite.

3. Stressed – Not Applicable and Not Applicable – Stressed

   When a user inserts an extra stressed vowel that is not present in the target speech, "Stressed – Not Applicable" occurs. When a user misses a vowel that is stressed in the target speech, "Not Applicable – Stressed" occurs.

4. Unstressed – Not Applicable and Not Applicable – Unstressed

   When a user inserts an extra unstressed vowel, which is not present in the target speech, "Unstressed – Not Applicable" occurs. When a user misses a vowel, which is unstressed in the target speech, "Not Applicable – Unstressed" occurs.

Ideally, we should report any unmatched results as stress errors. However, as our next step is to identify rhythm errors in a user's speech we should mainly consider stress errors that could cause incorrect speech rhythm.

As discussed in sub-section 2.2.2 (page 18), speech in English is widely recognised as having stress-timing rhythm. Stressed vowels construct the main frame of speech rhythm. It means that while reading a sentence in a given context, any missing or additional stressed vowel would destroy the expected rhythm. Contrarily, unstressed vowels play less important roles in speech rhythm. Native speakers sometimes skip unstressed vowels but increase the length of neighbouring consonants or short pauses to retain the rhythm. Therefore, for the above four kinds of situations, we must report the second kind of situation because cases in that are absolute stress errors. We also need to treat the third kind of situation seriously since a stressed vowel should never be unpronounced and an additional stressed vowel is not allowed in order to produce good speech rhythm. We tend to ignore the fourth kind of situation since it has less negative impact to speech rhythm.

5.4 Rhythm Error Identification

If no stress error is found, the error identifier then derives a rhythm pattern by extracting the stressed vowels from the stress pattern. Since a rhythm pattern consists of a sequence of stressed vowels and intervals between each pair of them, the system must retrieve additional timing information from the output of the speech recogniser to construct the rhythm pattern.

In order to identify rhythm errors in a user's speech, we developed a rhythm error identification approach. The approach is to compare the user's rhythm pattern with a target rhythm pattern to identify the differences. The differences are then further analysed in order to eliminate insignificant, minor, or redundant errors. By using this approach, we need to carefully choose a native speaker who can produce the right speech rhythm pattern so that the comparison results based on the target rhythm pattern will not lead to suggestions that might negatively affect the user. Because the user's speech may have a different length to the target speech, directly comparing the rhythm pattern of the two speeches may not always be practical. Therefore, we normalised the length of the user's speech to the length of the target speech.

We explored two kinds of comparison approaches for identifying the differences in the user's speech, *VOP comparison* and *foot comparison*.

5.4.1 VOP Comparison

When speaking rhythmically, English speakers adjust the overall timing so that vowel onsets of the stressed vowels occur near certain privileged temporal locations. Thus Vowel Onset Points (VOP) are believed to be the most important elements forming the English speech rhythm pattern [3, 53].

The VOP comparison approach only compares VOPs of two corresponding stressed vowels in the user's speech and the target speech. The VOP differences can indicate whether the user starts a stressed vowel earlier or later than the target. The advantage of using this method is that the comparison is fairly simple but the disadvantage is that any consequential "error" caused by previous errors will also be reported. For example, Figure 5.7 illustrates the rhythm pattern comparison between the target speech and the user's speech for a sentence "Amongst her friends she was considered beautiful.". The stars represent the stressed vowels in each utterance and the positions of those stars indicate the VOPs of those stressed vowels. From this figure, we can see that none of the VOPs in the user's speech match the corresponding VOPs in the target speech, especially from the third to the fifth although it is clear that the fourth and the fifth VOP differences are caused by the late occurrence of the third VOP.

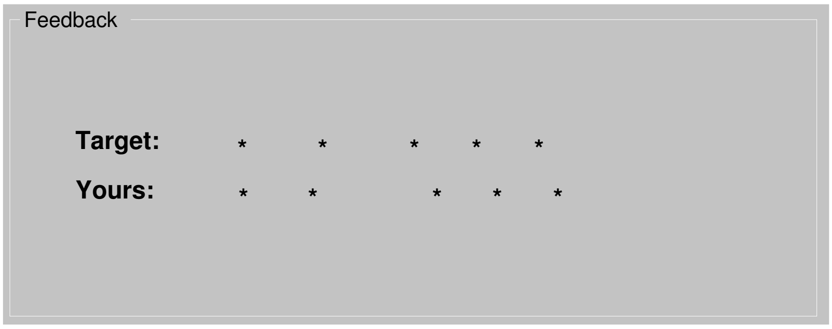


Figure 5.7: Rhythm pattern comparison – VOP.

By using this approach, all these five VOP differences are reported to Peco. Obviously it is really not ideal. Peco may be disturbed by those consequential errors and may give unnecessary, incorrect, or useless feedback to the user.

5.4.2 Foot Comparison

The foot is calculated by measuring the distance between VOPs of a pair of stressed vowels. The foot comparison approach compares a foot between two stressed vowels in the user's speech with the corresponding foot in the target speech. The foot differences can indicate whether the user holds a foot too long or too short. The advantage of this method is that, unlike the VOP method, foot comparison can ignore the consequential errors and only report the master errors. Thus Peco can focus on the master errors and provide the user right and handy feedback.

Foot difference can be measured in two forms. One is the absolute foot difference, which is directly calculated by subtracting the foot in the user's speech from the corresponding foot in the target speech. The other is the relative foot difference, which is the ratio of the absolute foot difference over the foot in the target speech. We use both absolute and relative foot differences to identify rhythm errors because in some circumstances the relative foot difference may reduce the limitation of the absolute foot difference in identifying the causes of a wrong rhythm. For example, an absolute foot difference in a user's speech may not be significant. But the foot may be over 200% longer or shorter than the corresponding target foot if the target foot itself is small.

The foot differences can be either positive or negative. By "positive", it means the foot in a user's speech is shorter. By "negative", it means the foot in a user's speech is longer.

Clearly, in acceptable utterances of a sentence spoken by different speakers or by the same speaker multiple times, none of corresponding feet will be exactly the same in different utterances. So we need to have an absolute difference threshold and a relative difference threshold to measure whether the foot differences are acceptable or should be considered as errors. In our prototype system, we adopt an interactive method: thresholds are not set to fixed numbers but are adjustable by users. Not only does this interactive method enable researchers to find appropriate thresholds during the use of the prototype software, but also it provides users an opportunity to know how closely their rhythm matches the target.

In order to let users concentrate on their main rhythm errors, our prototype system only reports the largest absolute foot difference and the largest relative foot difference to Peco. For example, for the case illustrated in Figure 5.7, with the threshold of the absolute foot difference set to 5 units and the threshold of the relative foot difference set to 20%, our system reports that in the user's speech the first foot is absolutly too short, the second foot is absolutly too long, and the second foot is also relatively too long.

5.5 Visualisation Tools

In our prototype system, we provided users with some visualisation tools. These visualisation tools can be thought as the preliminary design of Peco. Currently they are designed for experimenting by ESL researchers and for helping users get some information of their prosodic problems.

5.5.1 Stress Error Visualisation

The stress error visualisation tool shows users the result of their stress pattern comparison.

For example, in Figure 5.8, the target stress pattern is shown in the upper panel with the title "Target" and the user's stress pattern is shown in the lower panel with the title "Yours". From the figure, we can see that the stress statuses of vowels /ɪ/ and /ə/ in the user's speech are opposite to the ones in the target speech. In addition, there is one vowel /ɪ/ missing in the user's speech.

5.5.2 Rhythm Error Visualisation

The rhythm error visualisation tool shows users the results of their rhythm pattern analysis. Note that the feedback in the figures in this section are based on the absolute foot difference with an arbitrary threshold value.

As discussed in section 5.4, the rhythm error identification procedure is constrained by the results of the stress error identification. If either the

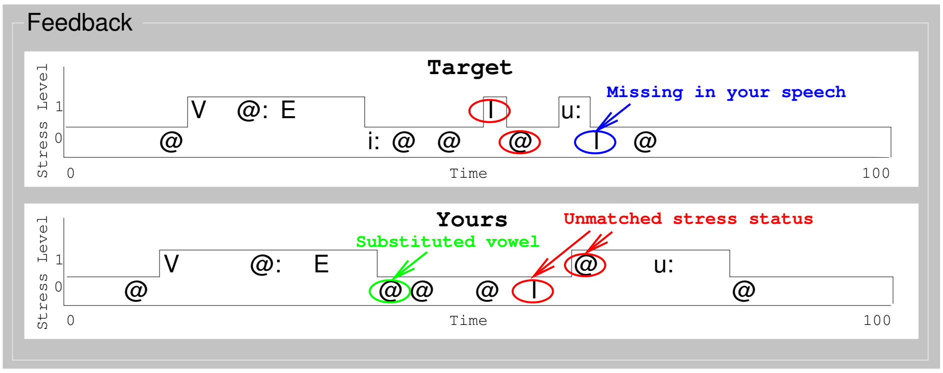
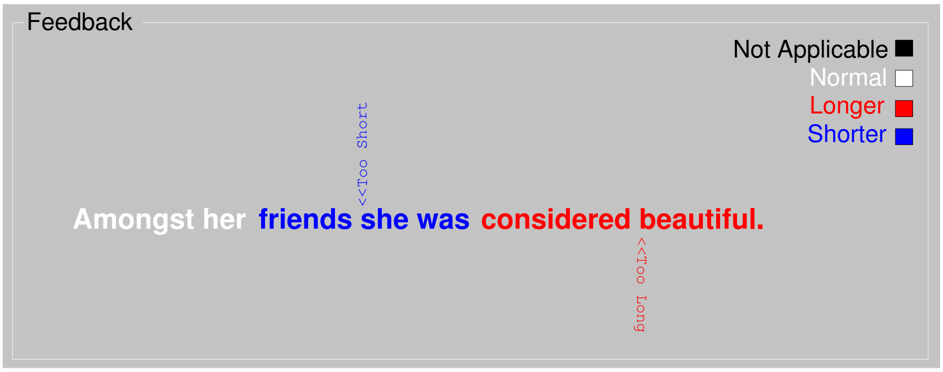


Figure 5.8: Stress pattern visualisation.

user makes any serious stress errors or our stress detector reports wrongly, which means some of the feet in the user's speech do not match with the target feet, then the foot comparison approach used for identifying the rhythm errors will not work appropriately. This implies that our originally designed rhythm error identification procedure may not be able to give users constructive feedback on rhythm as we would hope. Therefore, we provide two additional timing measurement options in order to help users identify their timing problems in their speech. Thus we have three options in total to provide users with three different kinds of feedback on timing and rhythm errors.

The first option is *Target Vowel to Vowel*, where the intervals between adjacent vowels in the target speech are compared with the intervals between the corresponding vowels in the user's speech, regardless of the stress status in the target speech. This option tends to check whether the overall speech rate in the user's speech matches the one in the target speech.

For example, Figure 5.9 shows the rhythm errors in the user's speech by using the target vowel to vowel option for an ESL user's utterance of the sentence "Amongst her friends she was considered beautiful", . The speech rate in words "Amongst her" is acceptable but in words "friends she was" is faster. The shortest interval is between the words "friends" and “she”. The overall speech rate in words “was considered beautiful” is slower than the target and the longest phase occurs between the last /ə/ of the word “considered” and the first vowel /uː/ of the word “beautiful”. According to this, the user should slow down somewhat at the beginning and possibly needs to add a short pause after the word “friends”. The user also needs to speed up a bit during the interval towards the end of the sentence.



Amongst her friends she was considered beautiful.

Figure 5.9: Timing feedback – target vowel to vowel.

The second option is *Target Stressed to Stressed*, where the intervals between stressed vowels in the target speech are compared with the intervals between the corresponding vowels in the user’s speech. Note that it does not count the unstressed vowels in the target speech and ignores the stress status of the user’s speech. This option can roughly discover whether the user has potentially presented a kind of “rhythm” despite there being serious stress errors in the user’s speech.

For the same example used above, Figure 5.10 shows the rhythm error in the user’s speech by using the target stressed to stressed option. It suggests that the user does have a kind of timing control while starting and finishing the reading but needs to adjust the timing by slowing down the middle part of the sentence.

Note that a vowel in the target speech will be skipped if there is not a

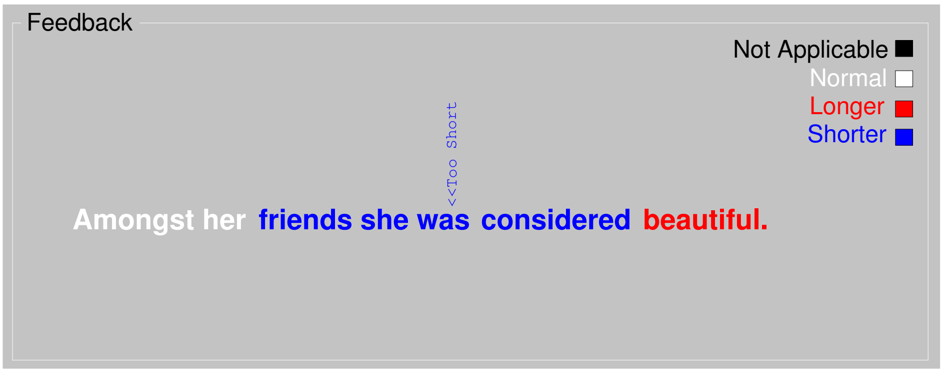


Figure 5.10: Timing feedback – target stressed to stressed.

corresponding vowel in the user's speech when the above two options are used.

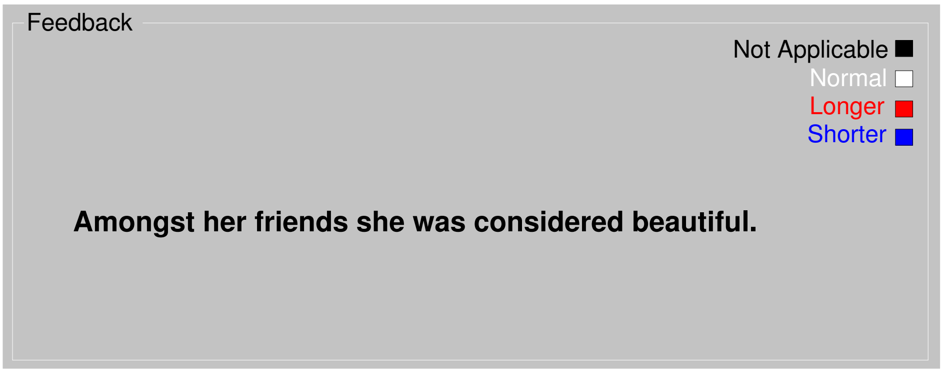
The third option is *Exactly Matched Stress Pattern*, where the intervals between the stressed vowels in the user's speech are compared with the intervals between the corresponding stressed vowels in the target speech. This applies if and only if the stressed vowels in the user's speech exactly match the stressed vowels in the target speech.

For the same example used above, by using the exactly matched stress pattern option, the system will display a warning and only give "not applicable" feedback as shown in Figure 5.11 due to the existing stress errors in the user's speech.

5.6 Chapter Summary

In this chapter, we have described the final component of Span — the error identifier, including the vowel sequence alignment algorithm, the stress error identification, and the rhythm error identification.

A two-layer vowel sequence alignment algorithm based on Needleman/Wunsch technique has been introduced to handle the increased difficulty and ambiguity while dealing with ESL speakers' speech.



Feedback

Not Applicable ■
Normal □
Longer 🟥
Shorter 🟦

Amongst her friends she was considered beautiful.

Figure 5.11: Timing feedback – exactly matched stress pattern.

Several kinds of mismatch between the stress status of vowels from a user's speech and the target speech were studied. Two rhythm error identification methods were explored.

We also presented the preliminary work on Peco, which provides users with visual feedback for their prosodic problems.

Chapter 6

Conclusions

The objective of this chapter is to summarise the research findings and present future work. Following an overview of this research, findings in each of the three main components of Span are explored. The thesis then concludes with a discussion of future research work.

6.1 Conclusions

The work in this thesis involved technologies in multiple areas, including computer science, linguistics, statistics, and physics. The purpose of this thesis was to use these technologies to develop a prototype sub-system — Span — of an ICAI system for TESOL. The goal was successfully achieved by constructing a speech recogniser, building a stress detector, and creating an error identifier.

The performance of the speech recogniser and the stress detection components of Span was reasonable (about 87% accuracy on the speech recogniser, and 85% accuracy on the stress classifier). This performance is comparable with the reported performance of other systems in related areas. The performance of the error identifier component could not be evaluated since there exists an unsolved lingustic research problem — foot difference thresholds.

It is not clear whether the performance of each of the components of Span is adequate for the effective use of the ICAI system because the overall performance of Span cannot be evaluated until the foot difference thresholds have been determined and the other sub-system — Peco — has been completed, which are beyond the scope of this thesis.

6.1.1 Speech Recognition

We studied an HMM–based forced alignment speech recogniser. The speech recogniser is a key component of Span. The central requirement on the speech recogniser is that it can accurately identify the boundaries of the phonemes in a speech signal.

We explored a range of parameters for constructing the speech recogniser, especially in the phoneme HMM stochastic design and the speech encoding process. Our findings supported Hypothesis 2.1.1 (page 17). We found that a stochastic model with 4 Gaussian mixtures per HMM state contributed to the highest vowel phoneme boundary accuracy on our speech data set. We found that 12 MFCCs plus 0th Cepstral, and their derivative and acceleration features computed for a 15 ms wide speech signal window with 11 ms interval provided sufficient information for the speech recogniser to minimise the boundary timing errors on the vowels on our speech data set. The maximum accuracy within the 20 ms boundary timing difference threshold was 87.07%, which is higher than other related speech recognition systems mentioned in section 2.4.1 (page 26). However, we must point out that a precise performance comparison is not possible due to the variations in the speech data used and details of the implementations.

We also looked at the standard three-state, left-to-right phoneme HMM architecture. We found that this architecture is a possible source of some of the errors of the recognised phoneme boundaries. The left-to-right, no skipping state connection model requires that every recognised phoneme must be at least three frames wide. If the actual phoneme is missing, or the duration is less than the width of three frames, then sound signals belonging to its neighbours are "stolen" to make up the phoneme in order to meet the requirement. This "stealing" problem generates bad phoneme boundaries.

6.1.2 Stress Detection

On advice from the linguistic researchers [77], we decided the rhythmic stress is more important in good speech production. We developed an approach to rhythmic stress detection in NZ English.

Vowel segments were identified from speech data and a range of prosodic features and vowel quality features were extracted from the vowel segments instead of from commonly used syllables. We explored several normalisation methods for normalising the prosodic features. We developed an innovative method for calculating vowel quality features.

We normalised and/or scaled different combinations of these features and then fed them into the C4.5 and LIBSVM algorithms to learn the stress detectors. We found that a combination of duration and amplitude features achieved the best performance (84.72%) and that the vowel quality features also achieved good results (82.50%). It is interesting to note that the prosodic features and the vowel quality features are comparable at detecting stress, but that their combination did not appear to enhance performance.

The experimental results supported Hypothesis 4.2.1 on page 55: features we used in our study are largely carried by the vowel as the nucleus of the syllable. The features extracted from vowels can significantly contribute to the rhythmic stress detection.

The results using vowel quality features supported Hypothesis 2.2.1 (page 22): for detecting stress status, knowing a vowel is reduced is more reliable than knowing it is full.

We found that on our data set, support vector machines achieved better results than decision trees, so we decided to use support vector machines to construct the stress detector.

We found that with the whole set of features, either scaled or unscaled, neither support vector machines nor decision trees could perform well at handling less useful features. However, with the smaller set of scaled vowel quality features, the support vector machines were able to deal with redundant features but decision trees still could not.

While the maximum accuracy is not good enough yet to be very useful for a commercial system, these results are quite comparable to (even slightly better than) other stress detection systems in this area [39, 72], reflecting the fact that automatic rhythmic stress detection from continuous speech remains a difficult problem in the current state of the art of speech recognition.

6.1.3 Error Identification

We explored the Needleman/Wunsch algorithm to align vowel sequences in the target speech and the user's speech. We found that due to too many variations and errors in ESL speakers' speech, the vowel sequence alignment process encountered many ambiguities and difficulties and produced more than one possible alignment result even though the program used additional consonant information. We designed and implemented a strategy using a two-layer alignment approach that greatly reduced the ambiguity in the vowel sequence alignment process.

We explored kinds of comparison results between the stress status of pairs of vowels from the target speech and the user's speech. We determined which kinds of mismatch are the most serious errors. Our system is able to distinguish errors and identify the most serious errors based on our exploration.

We also explored two rhythm error identification methods. We found that the VOP comparison approach was easily implemented but reported consequential errors to Peco in addition to the key errors. We found that the foot comparison approach was much better at reporting only the key errors. However, due to the variations of human speech, setting up the thresholds that are used to determine whether the foot differences should be considered as errors or as acceptable variations is still under investigation in the linguistic research area.

We found that, in practice, only giving feedback on rhythm errors was not very useful, because the prerequisite is that all the stressed vowels in a user's speech and in the target speech must be correctly matched. It was too hard to achieve this for two reasons: first, it is not easy for ESL speakers to produce matched stress status on all corresponding vowels without enough practice; and second, our speech recogniser and the stress detector did not provide 100% accuracy, which means even though a user's stress pattern actually matches the target, it may still be reported as having stress errors. Therefore, in addition to the rhythm error identification based on a strictly matched stress pattern, we also provided two other timing measurements — target vowel to vowel and target stressed to stressed — in our visualisation tools in order to help users to get more sense of their stress and rhythm problems. The usefulness of the two additional timing measurements needs to be further investigated by linguistic researchers.

6.2 Future Work

Possible improvements to the performance of Span are discussed in this section.

6.2.1 Improving the Forced Alignment System

The word pronunciation dictionary we use currently contains some alternative pronunciations of some words. These alternative pronunciations reflect variations in pronunciation that are acceptable to native speakers. Clearly, we need to continue augmenting the dictionary with other alternative pronunciations that are acceptable to native speakers. Although we augmented a small number of words in the dictionary for Mandarin speakers in a small experiment (see page 76), the dictionary does not include the common pronunciation mistakes of native speakers nor most of the mispronunciations of non-native speakers who are currently learning English. These mispronunciations cause auto-labelling errors in our system.

Figure 6.1 shows an example from a non-native speaker mispronouncing the word `pretend`. The dictionary pronunciation is /priˈtend/, but the speaker mispronounced it as /pˈtenə/ (she missed /ri/ and mispronounced the final phoneme /d/ as /ə/). The top viewport is the spectrogram of the sound of the word `pretend` mispronounced by the speaker; the middle viewport shows the hand-labelling of the sound waveform, and the bottom viewport shows the auto-labelling of the same sound waveform. Clearly, the boundaries of the auto-labelled phonemes have been badly affected by missing and mispronounced phonemes.

Missing certain phonemes (such as /r/) and adding phonemes such as /ə/ at word boundaries and within consonant sequences are common mistakes among ESL students, and our system needs to be able to deal with them more effectively. However, adding all the possible alternative pronunciations directly to the dictionary would make the dictionary very large, and would also greatly increase the complexity of the HMMs built by the forced alignment speech recogniser. This increased complexity would have unacceptable consequences for the computation speed of the recogniser.

We intend to build a model of the kinds of deletions, insertions, and substitutions that are common in the speech of Mandarin ESL speakers, and use this model to dynamically construct a better phoneme network that allows the recogniser to deal with insertion and deletion errors more

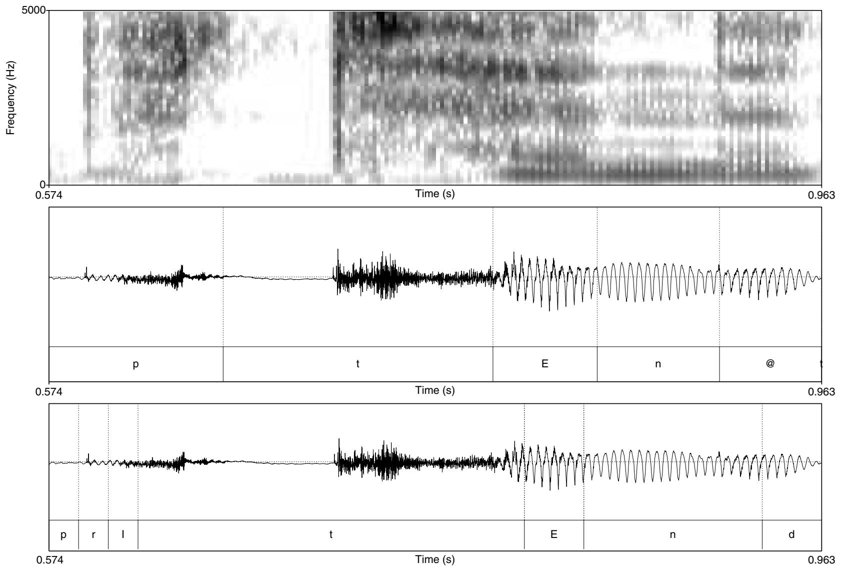


Figure 6.1: Auto-labelling errors of the forced alignment system.

gracefully.

6.2.2 Constructing A New Phoneme HMM Architecture

As pointed out in section 6.1.1, the standard three-state left-to-right phoneme HMM is also a possible source of phoneme boundary errors. We believe that the "stealing" problem can be addressed by a more robust HMM architecture in which some or all of the states can be skipped. This enhanced HMM is shown in Figure 6.2. With this enhanced HMM, we expect that the forced alignment based speech recogniser could handle missing phonemes by matching the phoneme against a zero length segment of the speech signal.

For long phonemes, particularly diphthongs, there may be more variations than can be captured well by just three states. We will consider HMMs with more than three states for such vowels.

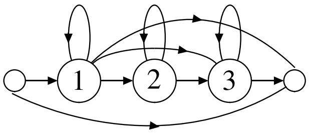


Figure 6.2: A new three-state HMM.

6.2.3 Adding A New Breathing HMM

In addition, the current HMM design deals with the short pause /sp/ and the silence /sil/ but does not model breathing properly since an audible breath has energy, and we do not have an HMM for breathing. Short periods of breathing currently confuse the recogniser, and cause it to label the phoneme boundaries badly.

Figure 6.3 shows an example of auto-labelling errors resulting from a short breath and an inserted phoneme. The first panel presents the spectrogram of a speech signal consisting of the word `but` preceded by a short breath and followed by an inserted /ə/, produced by a female ESL student. The second panel shows the sound waveform and the hand-labelling. The third panel presents the auto-labelling generated by the forced alignment system. As can be seen from the figure, the forced alignment system placed most of the boundaries quite inappropriately.

Audible breathing has less of an impact on the native speaker data as these speakers were largely able to read the sentences fluently on a single intake of breath. Non-native speakers usually have considerably more hesitation and therefore periods of breathing will be much more common. We will add an HMM model for breathing to our system to improve the system performance.

6.2.4 Others

There are a number of limitations in this work that need to be improved in the future.

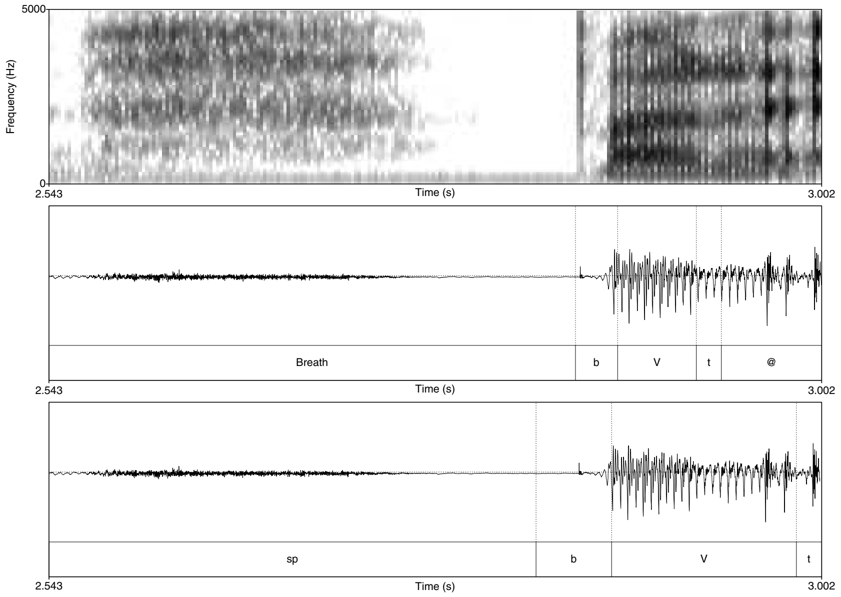


Figure 6.3: Auto-labelling errors with a breath and an inserted phoneme.

- We were surprised that the pitch features were not particularly useful for the automatic rhythmic stress detection. This is contrary to the discussions with linguistic researchers [77]. It might be possible that our pitch features were not sufficiently good, and/or that the pitch normalisation was not quite adequate for our data. In the future, we will examine better pitch detection and calculation algorithms, and investigate better normalisation methods.

- We suspect that the current duration and amplitude feature normalisation methods were not sufficient. We will explore other and better normalisation methods for rhythmic stress detection.

- Although we looked at several combinations of prosodic feature sets and vowel quality feature sets, we did not perform evaluations of each individual feature. Future work needs to do more exhaustive study of each of these features to identify which individual feature would contribute to better performance in rhythmic stress detection.

If it is too expensive to do an exhaustive search, we will explore the use of some appropriate feature selection algorithms.

- We used C4.5 and LIBSVM to construct the automatic rhythmic stress detector. We need to explore other techniques, including genetic programming and neural networks, as well as other DT constructors and other varieties of SVM.
- Once Peco has been completed, we will be able to evaluate the error identifier and further develop the stress and rhythm error identification methods to improve the whole system performance.

Bibliography

[1] ABERCROMBIE, D. *Elements of general phonetics*. Aldine Pub. Co., Chicago, 1967.

[2] ADAMS, C., AND MUNRO, R. R. In search of the acoustic correlates of stress: Fundamental frequency, amplitude, and duration in the connected utterance of some native and non-native speakers of English. *Phonetica 35* (1978), 125–156.

[3] ALLEN, G. The location of rhythmic stress beats in English: An experimental study. Part II. *Language and Speech 15* (1972), 179–195.

[4] ALTENBERG, B. Predicting text segmentation into tone-units. In *The London-Lund corpus of spoken English: description and research*, J. Svartvik, Ed. Lund University Press, 1990, pp. 275–286.

[5] AULL, A. M., AND ZUE, V. W. Lexical stress determination and its application to speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (1985), pp. 1549–1552.

[6] BEEFERMAN, S. The rhythm of lexical stress in prose. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics* (San Francisco, 1996), A. Joshi and M. Palmer, Eds., Morgan Kaufmann Publishers, pp. 302–309.

[7] BERNTHAL, J. E., AND BANKSON, N. W. *Articulation and phonological disorders*. Prentice Hall, New Jersey, 1988.

[8] BOCCHIERI, E., RICCARDI, G., AND ANANTHARAMAN, J. The 1994 AT&T ATIS chronus recognizer. Freely available on the Web at the page `http://www.research.att.com/~dsp3/slt95.ps`, 1994.

[9] BOERSMA, P. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In *Proceedings of the Institute of Phonetic Sciences* (Amsterdam, 1993), vol. 17, pp. 97–110.

[10] BOLINGER, D. Pitch accent and sentence rhythm. In *Forms of English: Accent, morpheme, order*, I. Abe and T. Kanekiyo, Eds. Harvard University Press, Cambridge, MA., 1965, pp. 139–180.

[11] BOLINGER, D. L. *Two kinds of vowels, two kinds of rhythm*. Indiana University Linguistics Club, Bloomington, 1981.

[12] CCITT. Recommendation G.711: Pulse code modulation (PCM) of voice frequencies, 1988.

[13] CHANG, C.-C., AND LIN, C.-J. LIBSVM: a library for support vector machines. Freely available on the Web at the page `http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf`, 2003.

[14] CHISTOVICH, L., VENTZOV, A., AND M., G. *The speech perception by human*. Nauka, Leningrad, 1976.

[15] COLE, R. A., MARIANI, J., USZKOREIT, H., ZAENEN, A., AND ZUE, V., Eds. *Survey of the State of the Art in Human Language Technology*. Cambridge University Press, 1996.

[16] CORTES, C., AND VAPNIK, V. Support-vector network. *Machine learning 20* (1995), 273–297.

[17] COUPER-KUHLEN, E. *An introduction to English prosody.* Edward Arnold, London, 1986.

[18] CRUTTENDEN, A. *Intonation*, second ed. Cambridge University Press, Cambridge, 1997.

[19] DALE, R. Tech858: Building spoken language dialog systems using VoiceXML. Freely available on the Web at the page `http://www.comp.mq.edu.au/units/tech858/Lectures/tech858-2002-03-12.pdf`, 2002.

[20] DAUER, R. Stress-timing and syllable-timing reanalyzed. *Journal of Phonetics 11* (1983), 51–62.

[21] DAVIS, K. H., BIDDULPH, R., AND BALASHEK, S. Automatic recognition of spoken digits. *Journal of the Acoustical Society of America 24(6)* (1952), 637–642.

[22] DAVIS, S. B., AND MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *IEEE transactions on Acoustics, Speech and Signal Processing* (1980), vol. 28, pp. 357–366.

[23] DEN OS, E. A., BOOGAART, T. I., BOVES, L., AND KLABBERS, E. The Dutch Polyphone Corpus. In *Proceedings of Eurospeech '95* (Madrid, 1995), pp. 825–828.

[24] ELLIS, D. ICSI Speech FAQ. Freely available on the Web at the page `http://www.icsi.berkeley.edu/speech/faq`, 2000.

[25] ESKENAZI, M. Detection of foreign speakers' pronunciation errors for second language training preliminary results. In *International Conference on Spoken Language Processing '96* (Philadelphia, PA, 1996), vol. 3, pp. 1465–1468.

[26] FISHER, D. H., PAZZANI, M., AND LANGLEY, P., Eds. *Concept Formation: Knowledge and Experience in Unsupervised Learning*. Morgan Kaufmann, 1991, ch. 1.

[27] FORGIE, J. W., AND FORGIE, C. D. Results Obtained From a Vowel Recognition Computer Program. *Journal of the Acoustical Society of America 31(11)* (1959), 1480–1489.

[28] FREIJ, G., FALLSIDE, F., HOEQUIST, C., AND NOLAN, F. Lexical stress estimation and phonological knowledge. *Computer Speech and Language 4*, 1 (1990), 1–15.

[29] FRY., D. B. Experiments in the perception of stress. *Language and Speech 1* (1958), 126–152.

[30] HARDCASTLE, W. J., AND LAVER, J., Eds. *The handbook of phonetic sciences*, first ed. Blackwell Publishers Ltd, Oxford, 1997.

[31] HENERY, R. Classification. In *Machine Learning, Neural and Statistical Classification*, D. Michie, D. Spiegelhalter, and C. Taylor, Eds. Ellis Horwood, 1994, ch. 2.

[32] HERMANSKY, H. Perceptual linear predictive (PLP) analysis for speech. In *Journal of the Acoustical Society of America* (1990), vol. 87(4), pp. 1738–1752.

[33] HOWITT, W. Linear Predictive Coding (LPC). Freely available on the Web at the page `http://www.otolith.com/otolith/olt/lpc.html`, 1995.

[34] HUNT, A. Comp Speech FAQ. Freely available on the Web at the page `http://www.speech.cs.cmu.edu/comp.speech/Section6/Q6.1.html`, 1997.

[35] HUNT, M. J. Signal Representation. In *Survey of the State of the Art in Human Language Technology*, R. A. Cole, J. Mariani, H. Uszkoreit,

A. Zaenen, and V. Zue, Eds. Cambridge University Press, 1996, ch. 1, p. 13.

[36] IRINO, T., MINAMI, Y., NAKATANI, T., TSUZAKI, M., AND TAGAWA, H. Evaluation of a speech recognition/generation method based on HMM and STRAIGHT. In *Proceedings of the International Conference on Spoken Language Processing* (Denver, 2002), vol. 4, pp. 2545–2548.

[37] ITAKURA, F. Minimum prediction residual applied to speech recognition. In *IEEE transactions on Acoustics, Speech and Signal Processing* (1975), vol. 23(1), pp. 67–72.

[38] JAIN, A. K., MAO, J., AND MOHIUDDIN, K. M. Artificial neural networks: A tutorial. *IEEE Computer 29*, 3 (1996), 31–44.

[39] JENKIN, K. L., AND SCORDILIS, M. S. Development and comparison of three syllable stress classifiers. In *Proceedings of the International Conference on Spoken Language Processing* (Philadelphia, USA, 1996), pp. 733–736.

[40] KAWAHARA, H., MASUDA-KATSUSE, I., AND DE CHEVEIGNE, A. Restructuring speech representations using a pitch adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication 27* (1999), 187–207.

[41] KOZA, J. R. *Genetic Programming — On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, 1992.

[42] LADEFOGED, P. *Three Areas of experimental phonetics*. Oxford University Press, London, 1967.

[43] LADEFOGED, P. *A Course in Phonetics*, third ed. Harcourt Brace Jovanovich, New York, 1993.

[44] Ladefoged, P., and Maddieson, I. Vowels of the world's languages. *Journal of Phonetics 18* (1990), 93–122.

[45] Liberman, M., and Prince, A. On stress and linguistic rhythm. *Linguistic Inquiry 8* (1977), 249–336.

[46] Lieberman, P. Some acoustic correlates of word stress in American English. *Journal of the Acoustical Society of America 32* (1960), 451–454.

[47] Lindgren, A. C., Johnson, M. T., and Povinelli, R. J. Speech recognition using reconstructed phase space features. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (2003).

[48] Makhoul, J. DSP Techniques. In *Survey of the State of the Art in Human Language Technology*, R. A. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue, Eds. Cambridge University Press, 1996, ch. 11, p. 402.

[49] Markel, J., and Gray, A. *Linear Prediction of Speech*. Springer-Verlag, Berlin, 1976.

[50] Mateescu, D. English phonetics and phonological theory. Freely available on the Web at the page `http://www.unibuc.ro/eBooks/filologie/mateescu`, 2003.

[51] Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. *Machine Learning, An Artificial Intelligence Approach*. Tioga Publishing Company, California, 1983.

[52] Mitchell, T. M. *Machine learning*. McGraw Hill, 1997.

[53] Morton, J., Marcus, S., and Frankish, C. Perceptual centers (P-Centers). *Journal of Pragmatics 83* (1976), 405–408.

[54] Muller, B., Reinhardt, J., and Strickland, M. T. *Neural Networks: An Introduction*, 2nd ed. Springer-Verlag, Berlin Heidelberg, Germany, 1995.

[55] NEEDLEMAN, S. B., AND WUNSCH, C. B. A general method applicable to the search for similarities in the amino acid sequences of two proteins. *Journal of Molecular Biology 48* (1970), 443–453.

[56] O'DELL, M. L., AND NIEMINEN, T. Coupled oscillator model of speech rhythm. In *Proceedings of The XIVth International Congress of Phonetic Sciences* (1999), vol. 2, pp. 1075–1078.

[57] PELLOM, B., AND HANSEN, J. Automatic segmentation of speech recorded in unknown noisy channel characteristics. *Speech Communication 25* (August 1998), 97–116.

[58] PELLOM, B. L. *Enhancement, Segmentation, and Synthesis of Speech with Application to Robust Speaker Recognition*. PhD thesis, Duke University, Durham, North Carolina, 1998.

[59] PENNINGTON, M. C. *Phonology in English language teaching: An international approach*. Longman, London, 1996.

[60] PIKE, K. L. *Phonetics: A critical analysis of phonetic theory and a technic for the practical description of sounds*. University of Michigan Press, Ann Arbor, MI., 1943.

[61] PIKE, K. L. *The intonation of American English*. University of Michigan Press, Ann Arbor, MI., 1945.

[62] QUINLAN, J. *C4.5: Programs for machine learning*. Morgan Kaufmann, 1993.

[63] RABINER, L. R., LEVINSON, S. E., ROSENBERG, A. E., AND WILPON, J. G. Speaker independent recognition of isolated words using clustering techniques. In *IEEE transactions on Acoustics, Speech and Signal Processing* (1979), vol. 27, pp. 336–349.

[64] RAPP, S. Automatic phonemic transcription and linguistic annotation from known text with hidden Markov models / an aligner for

German. In *Proceedings of ELSNET Goes East and IMACS Workshop* (Moscow, Russia, 1995), pp. 152–163.

[65] RIIS, S. K. *Hidden Markov Models and Neural Networks for Speech Recognition*. PhD thesis, Technical University of Denmark, 1998.

[66] ROACH, P. On the distinction between "stress-timed" and "syllable-timed" languages. In *Linguistic controversies*, D. Crystal, Ed. Edward Arnold, London, 1982.

[67] SAKOE, H., AND CHIBA, S. Dynamic programming algorithm optimization for spoken word recognition. In *IEEE transactions on Acoustics, Speech and Signal Processing* (1978), vol. 26(1), pp. 43–49.

[68] SJLANDER, K. An HMM-based system for automatic segmentation and alignment of speech. In *Procceedings of Fonetik* (2003), pp. 93–96.

[69] STUTTLE, M., AND GALES, M. A mixture of Gaussians front end for speech recognition. In *Proceedings Eurospeech* (2001).

[70] TEBELSKIS, J. *Speech Recognition using Neural Networks*. PhD thesis, Carnegie Mellon University, 1995.

[71] TSAI, M.-Y., CHOU, F.-C., AND LEE, L.-S. Improved pronunciation modeling by properly integrating better approaches for baseform generation, ranking and pruning. In *International Speech Communication Association Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation* (Estes Park, Colorado, USA, September 2002).

[72] VAN KUIJK, D., AND BOVES, L. Acoustic characteristics of lexical stress in continuous speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (Munich, Germany, 1999), vol. 3, pp. 1655–1658.

[73] VAN VUUREN, S. Comparison of text-independent speaker recognition methods on telephone speech with acoustic mismatch. In *International Conference on Spoken Language Processing* (1996), vol. 3, p. 1788.

[74] VELICHKO, V. M., AND ZAGORUYKO, N. G. Automatic recognition of 200 words. *International Journal of Man-Machine Studies 2* (1970), 223.

[75] WAIBEL, A. Recognition of lexical stress in a continuous speech system - a pattern recognition approach. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (Tokyo, Japan, 1986), pp. 2287–2290.

[76] WANG, X., AND POLS, L. C. A preliminary study about robust speech recognition for a robotics application. In *Proceedings of the Institute of Phonetic Sciences* (Amsterdam, 1997), pp. 11–20.

[77] WARREN, P., CRABBE, D., AND ELGORT, I., 2003. personal communication.

[78] WATSON, C. I., HARRINGTON, J., AND EVANS, Z. An Acoustic Comparison between New Zealand and Australian English Vowels. *Australian Journal of Linguistics 18(2)* (1998), 185–207.

[79] WIGHTMAN, C., AND TALKIN, D. The aligner: Text-to-speech alignment using Markov models. In *Progress in Speech Synthesis*, J. P. van Santen, R. W. Sproat, J. P. Olive, and J. Hirshberg, Eds. Springer-Verlag, New York, 1997, pp. 313–320.

[80] WIGHTMAN, C. W. *Automatic detection of prosodic constituents for parsing*. PhD thesis, Boston University, 1992.

[81] WITTEN, I. H., AND FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Diego, CA, 2000.

[82] XIE, H., ANDREAE, P., ZHANG, M., AND WARREN, P. Detecting stress in spoken English using decision trees and support vector machines. *Australian Computer Science Communications (Data Mining, CRPIT 32) 26* (January 2004), 145–150.

[83] XIE, H., ANDREAE, P., ZHANG, M., AND WARREN, P. Learning models for English speech recognition. *Australian Computer Science Communications (Computer Science, CRPIT 26) 26* (January 2004), 323–330.

[84] YING, G. S., JAMIESON, L. H., CHEN, R., MICHELL, C. D., AND LIU, H. Lexical stress detection on stress-minimal word pairs. In *Proceedings of the International Conference on Spoken Language Processing* (1996), pp. 1612–1615.

[85] YOUNG, S., EVERMANN, G., KERSHAW, D., MOORE, G., ODELL, J., OLLASON, D., VALTCHEV, V., AND WOODLAND, P. The HTK book (for HTK version 3.2). `http://htk.eng.cam.ac.uk/prot-docs/htk_book.shtml`, 2002.

[86] ZWICKER, E., FLOTTORP, G., AND STEVENS, S. Critical bandwidth in loudness summation. *Journal of the Acoustical Society of America 29* (1957), 548–557.

Appendix A

IPA Symbols and NZSED Labels

Consonants			Vowels		
IPA Symbol	NZSED Label	Sample Words	IPA Symbol	NZSED Label	Sample Words
/p/	p	pen, copy, happen	/ɪ/	I	kit, bid, hymn
/b/	b	back, bubble, job	/e/	E	dress, bed
/t/	t	tea, tight, button	/æ/	A	trap, bad
/d/	d	day, ladder, odd	/ɔ/	O	lot, odd, wash
/k/	k	key, cock, school	/ʌ/	V	strut, bud, love
/g/	g	get, giggle, ghost	/u/	U	foot, good, put
/ʧ/	tS	church, match, nature	/iː/	i:	fleece, sea, machine
/ʤ/	dZ	judge, age, soldier	/eɪ/	ei	face day, steak
/f/	f	fat, coffee, rough, physics	/aɪ/	ai	price, high, try
/v/	v	view, heavy, move	/ɔɪ/	oi	choice, boy
/θ/	T	thing, author, path	/uː/	u:	goose, two, blue
/ð/	D	this, other, smooth	/əu/	@u	goat, show, no
/s/	s	soon, cease, sister	/au/	au	mouth, now
/z/	z	zero, zone, roses, buzz	/iə/	i@	here, serious
/ʃ/	S	ship, sure, station	/eə/	e:	fair, various
/ʒ/	Z	pleasure, vision	/ɑː/	a:	start, father
/h/	h	hot, whole, behind	/ɔː/	o:	thought, law, north, war
/m/	m	more, hammer, sum	/uə/	u@	cure, jury
/n/	n	nice, know, funny, sun	/ɜː/	@:	nurse, stir
/ŋ/	N	ring, long	/ə/	@	about, coma, common
/l/	l	light, valley, feel			
/r/	r	right, arrange			
/j/	j	yet, use			
/w/	w	wet, one, when, queen			