Tone and INITIAL/FINAL Recognition for Mandarin Chinese
MSc Dissertation

John McKenna

September 13, 1996

Abstract

Chinese is a tonal language, with four main distinguishing tones. One school of thought argues that 'the tones are essentially independent of the vocal tract parameters of the syllables'. So the approach taken is to process tone and syllable (regardless of tone) in parallel. The other school argues that the correlation between excitation and vocal tract parameters is *not* negligible, and therefore the other approach's base syllable recognition rate will always be limited.

It is widely accepted that the tone contour in Mandarin syllables is affected by the initial consonant (INITIAL) , if there is one, and by the rest of the syllable (FINAL). In the early seventies Howie studied the domain of tone in Mandarin. Ho studied the effects of the syllable-initial consonant on tone. One aim of this dissertation project is to investigate whether the findings of Howie and Ho's studies can aid recognition of Mandarin, in particular the recognition of the tone.

Using a single speaker, and isolated monosyllabic words, various HMM (Hidden Markov Model) systems were designed and tested to see which could yield the best results and to see if this performed better than a simple 4-model system.

It was possible to achieve high accuracy rates by using 4 multimixture models for the FINAL)s. Using only F0- and energy- related information in the feature vectors performed slightly better than using F0-, energy, and MFCC- (Mel-frequency Cepstral Coefficient) information.

Dedication

This work is dedicated to the late Dr Michael Johnson - a gentleman and a scholar, who always helped and still inspires.

Acknowledgements

Many thanks to Dr Michael Johnson, and Dr Paul Taylor for their advice and encouragement; to Jing Fang, a most patient informant; to Stewart Smith for a smooth recording session; to all at the Department of Linguistics.

Contents

Chapter 1

Introduction

1.1 Historical review

Mandarin Chinese is a tone language. In a tone language, 'pitch has been captured for use in the phonemic system' (Ladd 1996), i.e. it functions to distinguish lexical items from one another.

It has been shown (Ho 1976), (Howie 1974), (Garding 1987), (Shih 1988) and (Shen 1990) that pitch height and slope are important parameters in distinguishing the tones. There are four tones whose contours are predictable, and a fifth, often called the neutral tone, whose pitch contour is less predictable.

Chinese uses a non-alphabetic character set, of which the characters number up to 50,000. The need to simplify input to machines has led to the development of over 200 methods, none of which compare in terms of efficiency with those used with alphabetic languages (ICC 1988).

Dictation machines offer prospects of such efficiency, and consequently much work on the design of such machines has been carried out over the past ten years. Over the past ten years, there have been varying approaches to the recognition of Mandarin Chinese speech.

The earlier ones chose to ignore the fifth tone. This greatly simplified the problem. In a practical system, the fifth tone cannot be ignored. Recent systems handle the fifth tone but use durational modelling to do so.

The predominant approaches that have been taken in the development of Mandarin lexical tone recognisers involve HMMs or Gaussian classifiers (GC).

Where HMMs are used, pitch-related information is almost invariably used as features in the observation vector. Energy-, and duration-related information is also used, and is particularly useful in classifying the 5th tone (Lee, Tseng, Gu, Liu, Chang, Hsieh & Chen 1990) and (Cheng, Sun & Chen 1990).

Tone 5 has always posed problems because it can be characterised by a short duration, and HMMs are unsuited to modelling duration unless duration is explicitly provided as a feature.

The Golden Mandarin (Lee, Tseng, Gu, Liu, Chang, Lin, Lee, Tu, Hsieh & Chen

1993), (Lee, Tseng, Chen, Hung, Lee, , Chien, Lee, Lyu, Wang, Wu, Lin, Gu, Nee, Liao, Yang, Chang & Yang 1993) and (Lyu, Chien, Hwang, Hsieh, Yang, Bai, Weng, Yang, Lin, Chen, Tseng & Lee 1995), and Tangerine (Hon, Yuan, Chow, Narayan & Lee 1994) and (Gao, Hon, Lin, Loudon, Yogananthan & Yuan 1995) systems have been at the forefront of Mandarin recognition. The developers of Tangerine have always favoured an integrated tone-syllable recognition approach, but at one point did use a separate GC for tone recognition. The Golden Mandarin series' designers prefer to identify tones independently of the base-syllable. Their latest system uses GCs similar to those used in an earlier version of Tangerine, but also uses disyllabic GCs to handle continuous utterances.

Of the four-tone systems, Yang, Lee, Chang & Wang (1988) cite the use of the statistical computation of mean, variation and range of pitch parameters to classify the four tones in speaker dependent recognition (Cheng & Sherwood 1992) and (Chan & Ng 1982). Yang et al. (1988) used Vector Quantisation (VQ) and Discrete HMMs (DHMM) to evaluate the effects of codebook size and tonal model topology. They achieved 98.33% recognition accuracy for the speaker-dependent case.

Cheng et al. (1990) implemented a multi-layer perceptron (MLP) in a speaker- independent tone recognition system. They divided the voiced part of the monosyllable into 3 segments. The energies, means, and slopes of the normalised pitch contour of each segment, plus the duration of the voiced part were used as the ten features in the input vector. It could be seen from these studies that training a neural network requires large amounts of training data and 'very long' training times (Owens 1993) which may make it unsuitable for speaker-adaptive systems, while training required for HMMs is considerably less.

Golden Mandarin I (Lee, Tseng, Gu, Liu, Chang, Lin, Lee, Tu, Hsieh & Chen 1993)[1], the 'first successfully implemented real-time Mandarin dictation machine', and Golden Mandarin II (Lee, Tseng, Chen, Hung, Lee, , Chien, Lee, Lyu, Wang, Wu, Lin, Gu, Nee, Liao, Yang, Chang & Yang 1993) were speaker dependent monosyllable recognition systems. 5-state DHMMs with 32-word VQ codebook were implemented for tone recognition. They gave a recognition accuracy of 95.5% for 5-tone recognition, which was slightly better than the 94.3% of the CHMM. In investigating CHMMs, it was found that a single mixture Gaussian density was adequate, since mixture densities were found to produce slightly lower or equal recognition rates.

Since then, more sophisticated techniques, using pitch analysers and a Gaussian Classifiers, have been used to develop speaker independent recognition systems that accept utterances continuous and have speaker-adaptive capabilities.

Hon et al. (1994) adopted an approach proposed by Lin, Lee & Ting (1993), who argue that the correlation between excitation and vocal tract parameters is *not* negligible and that systems that treat them as independent (e.g. Golden Mandarin) cannot achieve a high recognition rate. They claim that the errors in the separate tone and base-syllable recognition subsystems will multiply in the final recognition performance.

Lin, Lee & Ting used 10 cepstral coefficients and 10 corresponding delta cepstral co-

[1]Golden Mandarin I was first developed in 1990.

efficients (derived from LPC coefficients[2]) plus 1 normalised energy feature and 1 delta energy feature in a 22-feature vector. They implemented a 5-state CHMM[3] to simultaneously recognise tone and base-syllable. The result was a 94.85% tone recognition accuracy. They showed that it was possible to classify tones using feature vectors without pitch or excitation information. However they make no mention of the 5th tone.

Hon, Yuan, Chow, Narayan & Lee used a VQ stage with 4 codebooks[4] and 3-state DHMMs for both the initial consonant (INITIAL) and the remainder of the syllable (FINAL). Their best tone recognition accuracy was a 'mediocre' 86.4% and thus investigated other techniques.

Tangerine (Gao et al. 1995) finally addressed the 5th tone. In developing their speaker-dependent word-based system, they again used the integrated tone-syllable approach (Lin et al. 1993) and (Hon et al. 1994), this time using mel-scale frequency cepstral (MFCC) coefficients plus the pitch and its derivative as input to the DHMM. No specific recognition accuracy data was provided.

Fu, Lee & Clubb (1996) gives a good summary of more diverse techniques that have been applied. Most of the more recent approaches attain recognition accuracy rates above 95%.

1.2 Aims

There were two primary aims in this this project:

1. To examine how an integrated tone-syllable approach and a tone-independent-of-syllable approach compare.

2. To see if grouping segments together, based on similarities in their phonetic makeup, improved performance.

Having reviewed above the approaches taken, decisions were made accordingly on which approaches would prove most helpful and time saving.

It was hoped that the best recognition system would attain accuracy results above 90%.

The HTK toolkit was used to build the recognition systems. It is used to build continuous density hidden Markov model based recognisers.

1.3 Layout

This dissertation is organised as follows: Chapter 2 explains the nature of tone and syllable in Mandarin and the choice of vocabulary used in this study. The methodology is discussed

[2]See Rabiner & Schafer (1978) and Deller, Proakis & Hansen (1993).

[3]4 states were used for syllables without initial consonants.

[4]Codebook (CB) 1 quantised 12 LPC cepstral coefficients; CB 2 quantised 12 LPC delta cepstral coefficients; CB 3 quantised 12 LPC 2nd-order delta cepstral coefficients; CB 4 quantised power, delta power and 2nd-order power. CBs 1, 2, and 3 had 256 codewords and CB 4 had 32.

in Chapter 3. Chapter 4 looks at the technology involved in speech recognition, in particular HMMs and the HTK toolkit and uses the design of a recogniser to illustrate some of the issues involved. Chapter 5 explains the other approaches to tone recognition attempted in this study.

1.4 Transcription convention

It is appropriate now that the notation used hereafter is explained. The form of romanisation used throughout this paper is *pinyin.* However, *pinyin* is not completely phonetic. While the pronunciation of any syllable written in *pinyin* is distinct, sub-syllable segments written in *pinyin* may have more than one pronunciation. When referring to a complete syllable, the original *pinyin* transcription will be used. When referring to sub-syllabic units, they will be written using my own modified *pinyin* representation (MPR). MPR was used to label INITIAL and FINAL segments (See Section 3.2).

MPR is the same as *pinyin* except for:

1. Diacritics are not used. Instead the tone number is appended to the segment.
2. *ü* is written as `uu`.
3. *i* as in *shi* is written as `I`; *i* as in *qi* is written as `i`;
4. Initial-less syllables are written the same as the same segments appear when preceded by an initial, e.g. *wan* becomes `uan` , *yi* becomes `i`.

Chapter 2

Tone and Syllable in Mandarin

2.1 Introduction

The dialects of Chinese are all largely monosyllabic in the sense that the equivalence between morpheme and syllable is almost perfect. In Mandarin there are few morphemes of two syllables, and only one subsyllabic morpheme (the retroflex suffix, *r*). Constraints on the combination of segments in the syllable admit only 408 different segmental syllables (without the *r* suffix). MSC has 4 lexical tones and a neutral tone, resulting in a pool of 1632 possible combinations with which different syllables can be formed. In fact, due to lexical gaps, only 1,345 are actually used. This figure also includes syllables that carry the fifth tone. These are mainly grammatical particles which although occur quite often in speech, are few in number.

While the fifth tone is quite important, it has been ignored in this study. It is usually discriminated by duration, which was not employed as a parameter in this study.

2.2 Tone in Mandarin

The modification of tone has three acoustic dimensions: F0, intensity level, and duration. Of the three, F0 (perceived as relative pitch) is the most easily identified (Shen 1990).

The general tendencies of the four tones are commonly demonstrated on isolated tonic syllables. These citation forms practically never occur in live speech (Kratochvil 1968, 37), but they are useful as an illustrative starting point form the basis of all linguistic studies of tone.

These are illustrated in Figure 2.1 and described by Chao (1948, 24,25) as follows:

Tone	Chinese name	Description	Pitch	Diacritic
1	Yīnpíng Shēng	high-level	55	¯
2	Yángpíng Shēng	high-rising	35	´
3	Shǎng Shēng	low-dipping	214	ˇ
4	Qù Shēng	high-falling	51	`

The values in the `Pitch` column represent the pitch contour: `5` represents the highest of 5 levels; `1` represents the lowest. These values are relative and have to be adjusted for individual speakers and for style of speech. It should be noted that in non-final position in continuous speech, tone 3 undergoes tone sandhi. However, since this study only deals with isolated monosyllables, the sandhi forms are not dealt with.

2.3 The Syllable in Mandarin

The segmental breakdown of syllables is as follows (Howie 1974):

1. **Initial.** Initial consonant. Optional.
2. **Medial.** Non-syllabic vowel (*i* or *u*). Optional.
3. **Nucleus.** Syllabic vowel. Obligatory.
4. **Terminal.** Non-syllabic vowel or nasal consonant. Optional.

Wang (1967) and Chao (1968) say that the domain of tone is over the entire voiced portion of the syllable. Others argue that it is over segments 2-3 (Kratochvil 1968, 36) or only segment 3 (Dow 1972, 102). All of these are based on impressionistic observations. Howie's measurements (1974) suggest that the domain of tone be identified with segments 3 and 4, what is called the rhyming part.

Howie used 9 groups of monosyllables; the groups were classed according to the initial (or medial in syllables without initials).

The groupings were:

1. Those with initial syllabic vowel, e.g. *yi, a, wu, ai.*
2. Those with initial non-syllabic vowel, e.g. *ya, yao, you.*
3. Those with initial non-syllabic vowel and final nasal consonant, e.g. *yang, yuan, wen.*
4. Those with initial voiceless fricative, e.g. *shi, fa, xu.*
5. Those with initial voiced continuant, e.g. *ma, ni, lu.*
6. Those with initial aspirated stop, e.g. *pao, tu, ke.*
7. Those with initial unaspirated stop, e.g. *bi, du, ge.*
8. Those with initial aspirated affricate, e.g. *qie, can, chi.*
9. Those with initial unaspirated affricate, e.g. *jie, zan, zhi.*

He found that variations in the shapes of the F0 contours were associated with the segment preceding the syllabic vowel. The F0 contour seemed to move in an anticipatory manner during initial voiced consonants and non-syllabic vowels.

Ho (1976) examined the effects of syllable vowel and preceding consonant on tone. He found that, although the four contours maintain their characteristic shapes, different vowels and different preceding consonants do modify the four tone contours to varying degrees. The influence of the vowel is more considerable than that of the preceding consonant.

He showed that for the four tone contours, high vowels usually have higher fundamental frequency than mid-vowels, while mid-vowels usually have higher F0 than the low vowel. He showed that the F0 contours of V syllables differed from those of CV syllables. His comparison of contours of syllable nuclei after voiced and aspirated consonants showed that they start lower after voiced consonants than after voiceless consonants. Contours start lower after aspirated consonants than after unaspirated consonants.

2.4 Choice of vocabulary

The task of dealing with all the syllables of Mandarin is a large one. given the time constraints, it was decided to take a subset of the language to investigate the points in question.

The subset was based on groupings of segments similar to Howie's. The idea of grouping segments for recognition purposes is not altogether new[1].

The following groups were chosen for this study on the basis of there being a significant distinction between them in terms of their effects on tone as suggested by Howie's study. The abbreviated forms - shown in parentheses - are worth noting as it is used in later sections when referring to these groups.

1. Voiced continuant initial (`vc`): l, m, n
2. Voiceless fricative initial (`vlf`): f, h, s, sh, x
3. Aspirated affricate initial (`asaf`): c, ch, q
4. Monophthongs (`mon`): a, e, i, I, o, u, uu
5. i-medial diphthongs and triphthongs without final nasal (`iris`): ia, iao, ie, iu
6. u-medial diphthongs and triphthongs with final nasal (`urn`): uan, uang, un, ueng

All Oxford Concise Dictionary entries only for syllables (including all lexical tone variations) containing combinations of the above (a total of 268 toned syllables) were chosen as the language subset to be studied.

[1] Tangerine (Gao et al. 1995) classified segments according to their pronunciations and durations.

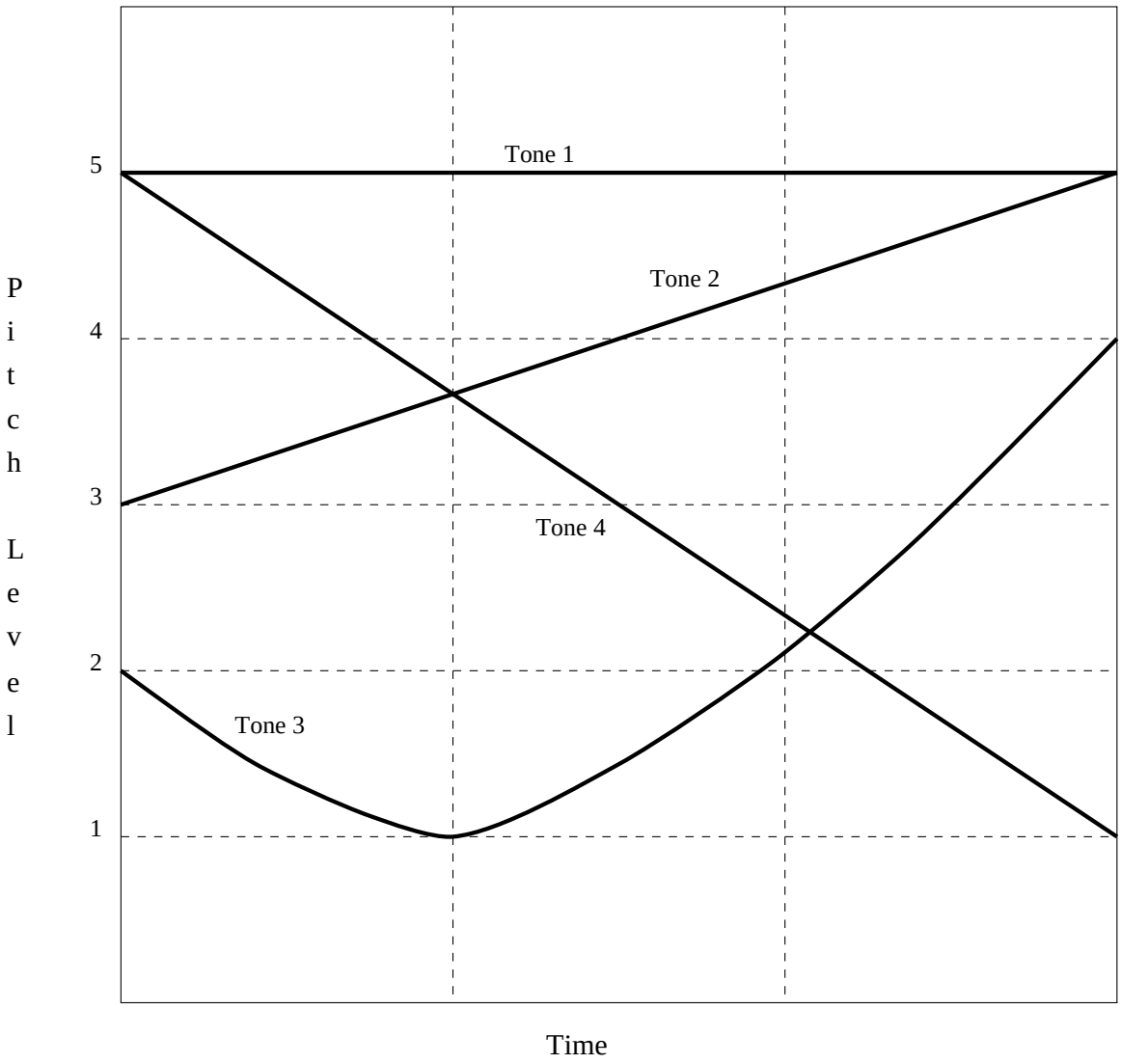

Figure 2.1: Idealised F0 contours of the 4 tones

Chapter 3

Methodology

3.1 Data Acquisition

One female Mandarin speaker was asked to provide all the training and test data. She was born and educated in Beijing and has lived in Edinburgh for two years.

Each syllable was written in *pinyin* including diacritics to mark the tone. This was preferred to using the ideograms, as some ideograms have more than one possible pronunciation. Ambiguity as to which pronunciation was intended was removed by using *pinyin*.

Six blocks were recorded. For each block, 268 syllables were randomised (each block differently). Five were to be used as training; one for testing. The main tone recognition systems cited in Section 1.1 generally use at least 5 training blocks. Such a small number is used because in a practical system, it is impractical to require a new speaker to produce too many training utterances (Lee, Tseng, Gu, Liu, Chang, Lin, Lee, Tu, Hsieh & Chen 1993). The idea is to try to utilise a limited, yet sufficient (see Section 4.2.4), number of training utterances efficiently.

The data was recorded to a digital audio tape (DAT) using the following recording equipment:

Microphone Sennheisser MKH815T RF condenser microphone

Mixing console Soundcraft 200b

Recording equipment Sony pcm2700A Digital Audio Tape recorder

Playback DAT Sony DTC60ES DAT recorder

The subject sat in a soundproof recording studio and read each syllable with an approximate frequency of one syllable every 2 seconds. The sheets from which they were read were placed on a table in front of the subject. The subjected was instructed not to touch the sheet, so as not to create noise. After each sheet was read, it was placed to one side, the next sheet was positioned, and the DAT was marked. After each block, the recording

was paused and the speaker allowed a brief rest and an opportunity to drink some water. This was intended to reduce the probability of laryngealising, which can occur after long periods of talking.

3.2 Labelling

Using the ESPS[1] program `xwaves` each syllable was delimited and viewed in the time domain, and in the frequency domain using a wideband spectrogram. Labelling was done using ESPS' `xlabel` label editor.

Since inconsistency in labelling is a potential source of error in training and recognition, it was necessary from the outset to decide on firm labelling criteria.

Two sources were consulted (Isard & Thompson 1996) and (Hieronymous, Alexander, Bennett, Cohen, Davies, Dalby, Laver, Barry, Fourcin & Wells 1990) and a combination of both plus some improvisation were used as is detailed below. The improvisation was primarily called upon when phenomena peculiar to Mandarin had to be dealt with. This was due to the fact that the HCRC and SCRIBE guidelines are intended for use with English.

Non-speech sounds

For non-speech sounds, the following labels were assigned:

sil For periods of silence between words (syllables).

#h Breaths in and out. Often occurring before words, or where the subject needed to catch her breath. Not all of these were labelled, as they were generally not visible in the time domain, and periods of silence were not aurally examined to detect non-silence phenomena. Thus, inevitably some breaths or sighs may have been included in segments labelled **sil.**

#s Lipsmacks and related lip or tongue clicks. If a something resembling a lipsmack was spotted, it was listened for. If it was not audible at normal volume level, it was ignored. If, on the other hand, it was audible, it was labelled. The labelled period was always made to be at least 36 ms (usually between 40 and 50 ms) so that at least one 25.6 ms parametrisation frame (see Section 3.3) 'captured' the label.

#n Noise. Usually inaudible (if very low frequency), this type of phenomenon was visible in segments of silence when viewed in the time domain.

[1] Entropic Signal Processing System

3.2.1 Initials

INITIALS were labelled in MPR (See Section 1.4) with tone number appended. Labelling depended on which category (See Section 2.4) they were part of.

vlf Voiceless fricatives: f, h, s, sh, x. The beginning was taken to be the point where frication, however slight, became visible on the spectrogram. The end was marked at where the onset of voicing of the following FINAL was seen.

asaf Aspirated affricates: c, ch, q. The HCRC guidelines for stop closures state:

> When post-silence stop-initial words occur, a period of silence roughly as long as the release should be included in the word. This is better than either nothing (bad for speech recogniser training) or a fixed amount of silence (not true to articulatory facts).

Affricates in English are generally unaspirated, so for the aspirated Mandarin ones, which are longer, it was decided to include a period of silence roughly half as long as the aspiration. Lipsmacks were often seen roughly half the length of the aspiration before the release burst (or onset of aspiration if there was no burst) supporting the decision taken - since lipsmacks often signal the beginning of a closure. If a lipsmack occurred at a another close point, this was instead taken to be the beginning of the closure.

vc Voiced continuants: l, m, n. The beginning was taken as the onset of voicing; the end was signalled by a sharp discontinuity in the formant frequencies and by an evident increase in the formant amplitudes of the following vowel.

3.2.2 Finals

These were marked by the onset and end of voicing, except for the two cases detailed:

Tails Tails are the the trailing low-amplitude noise at the end of a word. If following a voiced phone (all Mandarin syllables have a voiced final) it is usually characterised by an absence of voicing, but retains traces of the formants of the final phone, similar to a vowel- or nasal-sounding breath out.

Initially, tails were to be labelled, but for convenience and time-saving, the boundary between word and silence was set to the midway point of the tail. If the tail was very slight (low amplitude), it was excluded from the word completely. In retrospect, this was perhaps not the optimum choice. If included in the word, a HMM state could be assigned to it, however less than half the syllables had no tails. If included in the silence segment, it could corrupt the silence model unless an extra state or mixture was assigned. A third possibility would be to explicitly model tails. As long as short-time energy was a feature, in the parameter vector, it should avoid confusion

with vowel sounds. This would be reinforced by using F0 information as was done in part of this study. As it turned out, no major problems were encountered in adopting the midway criterion.

Post-silence clicks A throat or tongue click was often found before post-silence FINALS and *l*s. If the click resembled a lipsmack, it was labelled as such, unless there was less than one-and-a-half parametrisation frame lengths (40 ms) between the click and the syllable body, in which case it was included in the body.

3.3 Parametrisation

A C-shell script incorporating HTK's HCode was used to parametrise each data file. ESPS header information was removed before the parametrisation in to MFCCs (Mel-scale Frequency Cepstral Coefficients) was carried out. HCode allows the inclusion of arguments that generate delta MFCCs, energy (E), and delta energy (ΔE), if desired.

CSTR's SpeechTools[2] provided programs to generate F0, ΔF0 and probability of voicing (PV); PV having a value of either 0 or 1. The pitch tracker used a super resolution pitch determinator (SRPD) algorithm (Medan, Yair & Chazan 1991). Bagshaw, Miller & Jack (1993) in their study of PDAs (pitch determination algorithms) found that this algorithm performed well relative to the other PDAs studied.

For most tests, three types of feature vectors were used and the results of the tests compared. The three types were:

1. $C_1, \Delta C_1, E, \Delta E, F0, \Delta F0, PV$ (7 features)
2. $C_1..C_{12}, \Delta C_1..\Delta C_{12}, E, \Delta E$ (26 features)
3. $C_1..C_{12}, \Delta C_1..\Delta C_{12}, E, \Delta E, F0, \Delta F0, PV$ (29 features)

where C_n is the nth MFCC.

The three types were chosen on the basis of the following arguments:

3.3.1 F0-related information

F0 and $\Delta F0$ are useful acoustic quantities in recognising tone (See Section 1.1).

Lee et al. (1990) found that using only pitch frequency information was fine with 4 tones but poor with 5. Like Cheng et al. (1990), they used pitch frequency, short time energy and duration of the voiced part of the syllable as features, since short time energy is lower and the duration of the voiced part of the syllable is considerably shorter for the 5th tone.

So it would seem that the use of E, ΔE and duration would be helpful in being able to expand the system to handle the fifth tone. In this study, E, ΔE were used, but duration

[2]Centre for Speech Technology Research, University of Edinburgh.

was excluded as the four tones are relatively similar in duration for monosyllables (Ho 1976).C_1 and ΔC_1 were included as HCode requires at least one MFCC to calculate E; likewise for Δ MFCCs and ΔE.

3.3.2 MFCCs

MFCCs were used as they generally out perform other forms of parametrisation and tend to be the favoured parametric representation in most of today's speech recognition systems.

Tangerine (Gao et al. 1995) used MFCCs. Its forerunner (Hon et al. 1994) used LPC based speech representation. Using MFCCs was reported to have reduced recognition errors by 17%. However they also claim that although using MFCCs provides a good representation of speech, it is not very robust to environmental and acoustical disturbances, and so they recommend the use of spectral subtraction and cepstral mean normalisation to make the representation more robust.

MFCCs without these noise handling enhancements were considered satisfactory for the speech used in this study which was recorded in a controlled environment (See Section 3.1).

MFCCs are generated as follows:

If P is the analysis order, and N is the number of MFCCs required, then P triangular bandpass filters are equally spaced out along the mel-scale, where the mel-scale used by HTK is:

$$Mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{3.1}$$

The MFCCs are generated using the following discrete cosine transform:

$$c_i = \sum_{k=1}^{P} X_k \cos\left[i\left(k - \frac{1}{2}\right)\frac{\pi}{P}\right], \qquad i = 1, 2, \cdots, N \tag{3.2}$$

where $X_k, k = 1, 2, \cdots, P$, represents the spectral magnitude output of the kth filter.

A full discussion of MFCCs is beyond the scope of this dissertation. The HTK reference manual (Young 1992) gives details of the computation of the delta coefficients.

Young (1992) recommends an analysis order that is twice as large as the desired number of MFCCs. For our purposes, the analysis order was set to 24, as there were to be 12 MFCCs. HCode also generated E, ΔE and the Δ MFCCs.

The analysis frame period was set at 10 ms; the analysis window duration at 25.6 ms. These are the default settings in HCode. 2 windows were used to calculate the Δ MFCCs.

3.3.3 MFCCs and F0 Information

This feature vector was a combination of the two types of feature vectors above. How recognisers would perform using this combination was a question this project aimed to answer.

Chapter 4

HMMs and Recogniser Design

4.1 HMMs

A HMM is a 'stochastic finite state automaton' (Deller et al. 1993). It has been mainly used to model speech utterances, be they words, subword units, or phrases. They have also been used extensively, in one form or another, to model lexical tone in Mandarin[1]. HMMs attempt to characterise some of the variability in speech.

For a given frame, features of the entity to be modelled (in this case, tone) are extracted, parametrised, and form a feature vector. For tone, most cases (except Tangerine) the vectors include pitch-related information. A sequence of such vectors is called an observation string, $\mathbf{O} = \mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T$, where T is the number of observations in the string.

In the training phase, the HMM is taught the statistical make-up of the observation strings for its particular tone. The HMM is then capable of stochastically generating the tone it was taught.

In the recognition phase, given an input observation string, it is imagined that one of the trained HMMs generated the string. For each of the HMMs, the likelihood of that HMM generating the input string is computed. The tone that trained the HMM with the highest likelihood is deemed to be the input tone.

4.1.1 Model Parameters

A model $\mathbf{M}$ can be characterised by three sets of parameters:
$\pi = \{\pi_i\}$ or initial state distribution. π_i is the probability of state i generating the first observation O_1.
$\mathbf{A} = \{a_{ij}\}$ or state transition probability distribution. a_{ij} is the probability of a transition from one state i to another state j. Since there are an integral number of states, and hence an integral number of possible transitions, this is a discrete distribution.
$\mathbf{B} = \{b_j(k)\}$ or observation vector probability distribution. $b_j(k)$ is the probability of generating observation O_k from state j.

[1] Henceforth simply 'tone'.

At this time it is appropriate to explain the characterising difference between the discrete HMM (DHMM) and the continuous HMM (CHMM).

In DHMMs the number of observable vectors is reduced, usually by means of vector quantisation (VQ). In a VQ training session, neighbouring vectors are clustered about a codeword vector[2] which then acts as a representative of the cluster. The codewords together form a codebook. Generally the number of codewords in a codebook is 256 or less. This means that k in $b_j(k)$ above is an integral. Thus $\mathbf{B}$ has a *discrete* distribution for each state, with probabilities only assigned to the codewords. It should be noted that VQ is computationally expensive and, like any form of quantisation, can result in serious degradation.

Lee, Tseng, Gu, Liu, Chang, Lin, Lee, Tu, Hsieh & Chen (1993) found that for 4 tones, 8 codewords performed better than bigger codebooks, explaining that too many codewords may cause confusion. On the other hand 32 codewords was best for 5 tones.

Lee et al. (1990) concluded that DHMMs were best for 4 tones but that continuous HMMs (CHMM) and HMMs with bounded state durations performed better for 5 tones.

4.1.2 CHMMs

CHMMs avoid the quantisation problem. Since the observation vectors are continuous signals, a continuous probability density function (pdf) is used to characterise $b_j(k)$, which is better denoted by $b_j(\mathbf{O})$ (since k is often used for discrete values).

Generally, the pdf is of the form:

$$b_j(\mathbf{O}) = \sum_{\mathrm{m=1}}^{\mathrm{M}} \mathrm{c_{jm}}\, \eta[\mathbf{O}, \mu_{\mathrm{jm}}, \mathbf{C}_{\mathrm{jm}}] \qquad \text{for each state j} \tag{4.1}$$

where $\mathbf{O}$ is the vector being modelled, c_{jm} is the mixture coefficient for the mth mixture component in state j, and η is usually a Gaussian density function[3], with mean vector μ_{jm} and covariance matrix $\mathbf{C}_{\mathrm{jm}}$ for the mth mixture in state j.

A HMM state with a mixture density is equivalent to a multistate single mixture density model (Rabiner & Juang 1993). They say 'the distribution of the composite set of substates (each with a single density) is mathematically equivalent to the composite mixture density with a single state'. See Figure 4.2.

Mixtures offer an ability to capture variation within speech that would be treated as homogenous in the single mixture case. The difference between using single and double mixtures for inhomogenous data is illustrated in Figure 4.1.

Lee, Tseng, Gu, Liu, Chang, Lin, Lee, Tu, Hsieh & Chen (1993) found that a single mixture was adequate for 5 tones, in which case, Equation 4.1 becomes:

$$b_j(\mathbf{O}) = \eta[\mathbf{O}, \mu_{\mathrm{j}}, \mathbf{C}_{\mathrm{j}}] \qquad \text{for each state j} \tag{4.2}$$

If the η is a Gaussian distribution, then Equation 4.2 becomes:

[2]Usually the centroid of the cluster.

[3]The function can be 'any log-concave or elliptically symmetric density' (Rabiner 1989).

$$b_j(\mathbf{O}) = (2\pi)^{-\mathrm{n}/2} \left|\mathbf{C}_\mathrm{j}\right|^{-1/2} \exp[-(\mathbf{O} - \mu_\mathrm{j})^\mathrm{T} \mathbf{C}_\mathrm{j}^{-1} (\mathbf{O} - \mu_\mathrm{j})/2] \tag{4.3}$$

where n is the number of features in each vector.

4.2 Training and Recognition with WT7

The training and recognition of HMMs is best illustrated by example and also provides an opportunity to illustrate the issues raised so far. This will be done in the context of the implementation of a recogniser using the HTK toolkit.

The system in question is a 4-tone, 1 model per tone, using an F0 feature vector of seven elements (See Section 3.3). This kind of recogniser will be be referred to as WT7[4]. Each recogniser that is a subsequent variant (e.g. by virtue of having different numbers of mixtures) will be referred to as WT7_1, WT7_2, etc. This syntax will be used throughout this dissertation.

We will look at the following steps:

1. Data preparation
2. Training: initialisation
3. Training: reestimation
4. Recognition
5. Performance assessment
6. Discussion of performance

4.2.1 Data preparation

As was discussed in Section 3.2, the syllables were split into INITIAL/FINAL with each labelled in its modified pinyin representation (MPR).

For a 4-model tone recogniser, a word (syllable) is required to have one of four different labels; one label for each tone: `wt1, wt2, wt3, wt4` . HTK provides a label editor, HLEd, to replace, merge and delete (among other functions) labels.

To illustrate, for the syllable *shuàn* the labels of the INITIAL and FINAL are `sh2` and `uan2` respectively. These were then merged into a single label, `wt2` .

Implementing a HMM involves deciding on the type of model (ergodic, left-right), size (number of states), observation symbols (discrete, continuous), and in the CHMM case, the number of mixtures.

[4]WT denotes word (syllable) tone; 7 is the feature vector size.

There is no known simple, theoretically correct way of making these decisions; they are made depending on the signal being modelled. Since the HTK toolkit only uses CHMMs, they will be used exclusively.

A typical HMM for tone recognition has between 3 and 5 states. A left-right 4-state double-transition model is shown in Fig. 1. A left-right model is better suited to modelling signals whose properties change with time in a successive manner. Generally the decision as to the number of states to be used is made on the basis of empirical testing, and computational and performance considerations. A greater number of states does not imply better performance. Restricting the number of allowed transitions from a state ensures that large changes in state indices do not occur and reduces computational overhead.

A complex model may not be justified if the sequences associated with the model are not long or not rich enough (Rabiner & Juang 1993). Choosing too many states or mixtures would be in appropriate because it 'forces the use of an underspecified system'. As is mentioned below, the number of states is more flexible. When multimixtures are to be investigated in this study. A starting point of one mixture per state (mps) per model is used; the number of mixtures are then incremented one by one.

It is best to try to correspond the number of states to changing properties of the signal. With tone, this could be the number of segments between turning points in the pitch contour. A typical pitch contour for tone has two or three turning points (Yang et al. 1988). Hence the suitability of 3 or 4 states per model. It should be noted that one weakness of conventional HMMs is their durational modelling. This may cause recognition errors if the number of states is just enough to model each different pitch segment. Adding extra states aids the discrimination of tones which differ in length. Duration plays a dominant role in distinguishing the 5th tone. Explicit durational density models are preferable (Rabiner 1989), but complicate and increase computation.

Yang et al. (1988) studied the effects of tonal topology using VQ and DHMMs. They investigated 4 different types of models:

1. 3-node, single transition
2. 3-node, double transition allowed
3. 4-node, single transition
4. 4-node, double transition allowed (See Figure 4.1)

They found that all the above model topologies have almost equal performance when F0 features for the voiced portion of syllables are used.

Golden Mandarin I (Lee, Tseng, Gu, Liu, Chang, Lin, Lee, Tu, Hsieh & Chen 1993)[5], the 'first successfully implemented real-time Mandarin dictation machine', and Golden Mandarin II (Lee, Tseng, Chen, Hung, Lee, , Chien, Lee, Lyu, Wang, Wu, Lin, Gu, Nee, Liao, Yang, Chang & Yang 1993) were speaker dependent monosyllable recognition

[5]Golden Mandarin I was first developed in 1990.

systems.found that a single mixture Gaussian density was adequate, since mixture densities were found to produce slightly lower or equal recognition rates.

On this basis, a 4-node, double-transition-allowed model with one mixture per state (mps) was chosen as a starting point.

4.2.2 Training: Initialisation

The training of a HMM is the most difficult problem. This involves taking the initial estimate of the model generated by HTK's HInit, and then reestimating the model on presenting it with the training data.

For a given training sequence, the probability of the observation given the model, $P(\mathbf{O}|\mathbf{M})$, is generally a non-linear function of the parameters that the model has. This function will have local maxima in the multidimensional space. The optimal model corresponds to the global maximum of the likelihood function.

The problem here is finding initial estimates of the HMM parameters so that the local maximum is equal to or as near as possible to the global maximum of the likelihood function.

It is common practice to run reestimation algorithms a number of times with different initial estimates and to take the model that produces the highest local maximum as being the best model. However this has the disadvantage of added computational overhead.

Rabiner & Juang (1993) say that either random or uniform initial estimates of π and $\mathbf{A}$ are adequate in almost all cases. However, good initial estimates are essential for training CHMMs [6]. Rabiner (1989, 284) cites a number of ways to obtain good initial estimates. He suggests using manual segmentation techniques or segmental K-Means segmentation with clustering. It is a form of the latter that is used by HInit.

HInit iteratively calculates an initial set of parameter values using a segmental K-means procedure and Viterbi allignment. In the case of multiple mixtures, a modified K-Means clustering algorithm is used.

On the first iteration, the training data is uniformly segmented (according to the number of states). From this data, means and variances are calculated for each state.

On each reiteration, Viterbi allignment replaces the uniform segmentation. This finds the optimum state sequence through each observation sequence using the Viterbi algorithm (See Section 4.2). Here, 'recognition' is carried out on each observation sequence. Each observation vector is then assigned to the state that generated it by using the optimal path criterion and backtracking. Then means and variances are recalculated for each state.

This produces a maximum likelihood estimate for $\mathbf{B}$. The values of $\mathbf{A}$ remain unchanged and π is fixed by forcing every state sequence to begin in the first state.

When there are more than one mixture per state, the observation vectors assigned to each state must be divided into subsets (1 per mixture), also called clusters. This is usually done using a modified K-means clustering algorithm[7]. The clustering algorithm

[6] Good initial estimates are optional for training DHMMs

[7] A good account of the K-means algorithm is given in Deller et al. (1993, 71)

used by HInit[8] is 'rather crude (but fast) and convergence is not guaranteed when multiple mixtures are used'. As will be seen later, sometimes clustering cannot be achieved.

Before carrying out the initialisation process, it is necessary to provide HInit with a prototype HMM. The prototype defines the topology and characteristics of the HMM. A PERL script was written that generated the required prototype, when inputted with the number of states, number of mixtures, and observation vector size. Rabiner & Juang (1993, 380) say that 'it has been found that it is more convenient and sometimes preferable to use diagonal matrices'. The reason they give is the difficulty in reliably reestimating the off-diagonal components of the covariance matrix from limited training data. Thus, diagonal matrices were used on all occasions.

4.2.3 Training: Reestimation

Having obtained initial parameter values using HInit, the next step is to train the models. This is done using HRest.

For every training sequence, the parameters of a new model are reestimated from those of the old model until there is an improved model of the training sequence. At each iteration, the new model (if an improvement on the old) replaces the old model and another reestimation of the model is carried out until convergence is attained, or a specified maximum number of iterations has been executed.

The model can be improved by maximising π, $\mathbf{A}$, and $\mathbf{B}$ separately (subject to some stochastic constraints). If, after reestimation, the new set of parameters are a better set, it is retained, incorporated into the model, and the older set discarded.

The most popular method for locally maximising a model is the the forward-backward algorithm[9] This algorithm calculates the probability of an observation string being generated by a model in a computationally efficient way. HTK uses this algorithm for training when HRest is called.

A full discussion of the forward-backward algorithm is beyond the scope of this paper. However, a summary of how each set of parameters is reestimated (without derivation or explicit formulae) is provided.

$$\bar{\pi} = \text{expected number of times in state } i$$

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

$$\bar{b}_j(\mathbf{O}) = \frac{\text{expected number of times in state } j \text{ and observing } \mathbf{O}}{\text{expected number of times in state } j}$$

The left-hand sides represent the reestimates; the right-hand sides are calculated using the older values of the variables. The computation of the right-hand side is done using a

[8] Details are given in Young (1992, 86)

[9] Also known as the Baum-Welch reestimation algorithm.

set of forward and backward probabilities which can be calculated in a computationally efficient way.[10]

For the multimixture case c_{jm}, μjm, and $\mathbf{C}_{\mathrm{jm}}$ all need to be reestimated:

$$\bar{c}_{jm} = \frac{\text{expected no. of times in state } j \text{ with the } m\text{th mixture generating the observation}}{\text{number of times in state } j}$$

$\bar{\mu}_{jm}$, and $\bar{\mathbf{C}}_{jm}$ are weighted time averages of the observation vectors, having been weighted according to the likelihood of their being produced by mixture m in state j.

4.2.4 Recognition

Having trained up the models, we are ready to use them for recognition purposes. this is done using HVite. HVite allows the use of a grammar and bigram probabilities.

HVite takes a set of speech files and assigns labels to them using the trained models to make the assignment. The assignment algorithm used is the Viterbi algorithm. This is a dynamic-programming based algorithm which finds the single best state sequence. The best state sequence can be obtained by always locally choosing the best state sequence.

It is initiated by finding the probabilities of each state generating the very first observation:

$$\phi_1(j) = \pi_j b_j(\mathbf{o}_1)$$

Then, for each pair of time frames, a calculation is made of the probability of each state generating the first observation (of the pair), *and* of moving to each other state (or remaining in the same state), *and* of that next state generating the second observation (of the pair); the best local path being chosen:

$$\phi_{t+1}(j) = \max_{i=1,2,\cdots,N}\{\phi_t(i)a_{ij}\}b_j(\mathbf{o}_{t+1})$$

A record of the best local can be kept. The algorithm is terminated at the final observation. The Viterbi probability is then:

$$P^V = \max_{j=1,2,\cdots,N}\{\phi_T(j)\}$$

The model with the highest P^V is deemed to be the best match.

A grammar may be used to constrain the recognition sequence. This was used, but its discussion will be postponed for now.

Bigram probabilities can also be used, but for our purposes they were ignored. This was because it was desirable to have all tones equally likely.

[10]The forward probability $\alpha_i(t)$ is the probability of the partial observation sequence $\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t$, and being in state i at time t, calculated using an 'any path' criterion. The backward probability $\beta_i(t+1)$ is the probability of being in state j at time $t+1$ and of the partial observation sequence $\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, \cdots, \mathbf{o}_T$; again, using any path.

```
------------------- Overall Results -------------------
STATS: %Corr=81.27, Acc=29.59 [H=217, D=5, S=45, I=138, N=267]
-------------------------------------------------------
```

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	43	3	6	9	3	70.5
wt2	0	51	6	2	0	86.4
wt3	0	1	65	0	0	98.5
wt4	5	6	7	58	1	76.3
Ins	8	35	37	53		

Table 4.1: Results for Test WT7_1

Finally there are a couple of other issues relevant to tone recognition worth mentioning.

A single observation string cannot be used to train left-right models. A particular state will only be occupied for a small number of observations before a transition is made. So to make reliable estimates of the model parameters, multiple observation strings are required. The tone recognition systems cited in the overview generally use at least 5 observation strings.

4.2.5 Performance Assessment

Having carried out the recognition (the product being a series of recognised label files[11]), the next task was to assess the recognition performance.

Given that the test data had been hand-labelled as with the training data, these 'correct' labels are used as reference labels with which to compare the test labels.

HTK's HResults does this using a dynamic programming implementation. The output gives details of the number of correct (C), deleted (D), substituted (S), and inserted (I) labels, as well as the total number of labels in the reference label files. The output looks like Table 4.1.

The % correct and % accuracy figures are calculated from:

$$\%Corr = \frac{H}{N} \times 100$$

$$\%Acc = \frac{H - I}{N} \times 100$$

The figures in the far right column give the correctness of the individual labels.

[11] Henceforth referred to as the test labels.

It should be noted from the above results that the numbers of insertions and deletions in the *STATS* section do not tally exactly with the figures in the confusion matrix. There is an element of erroneousness in the results output by HResults. This will be discussed later.

Before going further into a discussion of the result just obtained, it is now appropriate to address one issue that has so far been ignored - the issue of non-speech modelling.

4.2.6 Non-speech Modelling

One approach to handling non-speech sounds is to ignore them by marking the word endpoints and placing each word in a separate file. This way the non-speech sounds are discarded and do not have to be modelled.

Another approach is to intentionally include non-speech at both ends of each training sample. The non-speech would be incorporated in the word models (most likely in the initial and final states).

The third approach, which was taken here, is to separately train explicit non-speech models.

It was found that 1 state, 2 mps models adequately modelled each non-speech sound, but only with the help of the HVite's grammar.

In our preliminary test, a grammar was used with the following syntax:

`NONSPEECH = sil` *or* `#h` *or* `#s` *or* `#n;`

`WORD = wt1` *or* `wt2` *or* `wt3` *or* `wt4;`

`PHRASE =` *any no. of* `NONSPEECH`*s followed by 1* `WORD` *followed by at least 1* `NONSPEECH;`

`SENTENCE =` *any number of* `PHRASE`*s* .

It was noticed that there were an alarming number of insertions. These were due to a breath or lipsmack (`#h` or `#s`) being inserted in the middle of a large number of words. These insertions did not directly account for the high number of insertions, as HResults was instructed to ignore non-speech sounds when assessing performance. However it caused single words to be split in two (with a `#h` or `#s` in the boundary between the two parts) See Fig 4.3. These extra 'words' caused the high number of insertions.

This type of insertion occurred mainly in `iris`- and `urn`-type syllables roughly where the transition was made from the non-syllabic *i* or *u* into the main vowel (See figure 4.3).

There were three solutions that were considered:

1. Employ durational information.

2. Improve the non-speech modelling.

3. Use the grammar to prevent the mid-word insertion of non-speech.

——————————————- Overall Results ——————————————-
STATS: %Corr=72.28, Acc=68.54 [H=193, D=10, S=64, I=10, N=267]
——-

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	37	1	1	20	5	62.7
wt2	0	52	5	1	1	89.7
wt3	0	2	64	0	0	97.0
wt4	4	23	7	40	3	54.1
Ins	0	4	1	5		

Table 4.2: Results for Test WT7_2

Option 1 offered the best prospects, however the omission of the 5th tone from this study was done to avoid the complications of including durational information.

Optimising non-speech modelling would distract from the aims of the project, so the third option was chosen.

Tone 3 syllables were particularly susceptible to this mid-word non-speech insertion. Han & Kim (1974) in their study of Vietnamese syllables found that the characteristic of the low-dipping 'broken tone' was heavy laryngealisation causing an abrupt dip that 'broke' the F0 contour and that creaky voice was heard during this dip.

In an attempt to prevent non-speech insertions, the grammar was modified so that a minimum number of non-speech segments had to appear consecutively between each pair of words. Using trial and error, it was found that a minimum of forty consecutive non-speech segments prevented all mid-word non-speech insertions.

When the affected words were examined, they were all found to be tone 3 and sounded creaky. Figure 4.4 shows this phenomenon when a grammar forced at least 15 consecutive non-speech segments between words. The 10 ms analysis window is large enough to capture the excitation cycles, but takes them for lipsmacks (See Figure 4.5).

The above figure of forty was equivalent to setting a lower limit of 415.6 ms on the length of inter-word periods of non-speech.

The results with this new constraint imposed are shown in Table 4.2.

Apart from the obvious improvement in accuracy, the figures of note here are those of deletions and insertions. They are curiously the same.

The reason for this is the dynamic programming algorithm and scoring system employed by HResults. In calculating the lowest cost path, substitutions have a penalty of 10 each; insertions have a penalty of 7 each. Sometimes an insertion and deletion cost less than two substitutions and therefore the allignment from which the results are calculated produce erroneous figures. The correct figures would appear as below:

——————————————— Overall Results ———————————————
STATS: %Corr=72.28, Acc=72.28 [H=193, D=0, S=84, I=0, N=267]
——

Actual results for Test WT7_2

As long as we are dealing with word (syllable) units, separated by longer periods of silence and constrained by our grammar, we should expect to have a one-to-one mapping between the reference labels and the test labels, unless the number of deletions and insertions are not the same. If they are the same, they can be added together, the total added to the number of substitutions, and the figures for deletions and insertions set to zero (as above). The accuracy figure becomes that of the correctness.

In different circumstances, an equal number of deletions and insertions should not be interpreted as misallignment. Although it is suggested that the figures be treated with suspicion, the only way to get truly reliable figures is to manually check the allignment of the transcription and test label files.

Similarly, the confusion matrices will only be a rough guide, not an accurate one.

This concludes the design of a complete recognition system. We will now look at a multimixture version of the same.

4.3 Multimixture WT systems

4.3.1 More WT7 tests

A prototype model was created with 4 states and 2 mps. As before, the models were initialised, reestimated and recognition performed.

The results are shown in Table 4.3. It should be noted that the figure of 42 insertions of `wt4` is another instance of the erroneousness that can appear in these confusion matrices.

With an accuracy of 94.76%, there is a marked improvement on the single mixture recogniser. Most of the errors arise from `wt3` being recognised as `wt4`.

The relatively poor accuracy rate in the single mixture case might be expected as all types of syllables are bundled together. For example, *shuǎn* has an initial period of voicelessness and by Howie's reckoning would only have a characteristic tone 3 contour (falling-rising) over the *an* part of the syllable. On the other hand *ǎ* has a characteristic tone 3 contour over the whole syllable. this does not make for a very robust model.

Increasing the number of mixtures will allow for some of the variability to be accounted for. Rabiner & Juang (1993, 377) says that 'lumping together all the variability from inhomogenous data sources leads to unnecessarily complex models, often yielding lower modelling accuracy'. This argument was not intended to support the idea of 'the more mixtures the better', but was directed towards using more models to cater for the different data sources. This will be done in Chapter 5. For now, we will further investigate the effects of mixtures.

——————————- Overall Results ——————————-
STATS: %Corr=94.76, Acc=94.76 [H=253, D=0, S=14, I=0, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	64	0	0	1	0	98.5
wt2	0	56	3	0	0	94.9
wt3	0	0	56	10	0	84.8
wt4	0	0	0	77	0	100
Ins	0	0	0	42		

Table 4.3: Results for Test WT7_3

——————————- Overall Results ——————————-
STATS: %Corr=76.40, Acc=76.40 [H=204, D=0, S=63, I=0, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	64	0	0	1	0	98.5
wt2	0	59	0	0	0	100
wt3	0	3	4	59	0	6.1
wt4	0	0	0	77	0	100
Ins	0	0	0	42		

Table 4.4: Results for Test WT7_4

Looking at the results of WT7_2 and WT7_3, it seems that the single mixture model for `wt3` performs better. A test was performed using a single mixture model for `wt3` and double mixture model for the other three tones. The results are shown in Table 4.4.

The majority of `wt3`s were 'claimed' by `wt4`. This shows three things:

1. The performance of an individual model is not independent of the other 'competing' models. In WT7_3 `wt3` has a correctness figure of 84.8%; in WT7_4, the figure is 6.1%.

2. A model that has a more accurate representation (in this case `wt4`) of the variability of its data source may claim tokens belonging to a model with a less accurate representation (in this case `wt3`) of the variability of its data source.

——————————- Overall Results ——————————-
STATS: %Corr=71.16, Acc=70.04 [H=190, D=3, S=74, I=3, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	0	26	0	36	3	0.0
wt2	0	58	1	0	0	98.3
wt3	0	0	55	11	0	83.3
wt4	0	0	0	77	0	100
Ins	0	1	1	1		

Table 4.5: Results for Test WT7_5

——————————- Overall Results ——————————-
STATS: %Corr=85.39, Acc=83.52 [H=228, D=5, S=34, I=5, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	64	0	0	1	0	98.5
wt2	6	28	19	1	5	51.9
wt3	0	0	59	7	0	89.4
wt4	0	0	0	77	0	100
Ins	0	0	1	3		

Table 4.6: Results for Test WT7_6

3. These results reinforce the idea that HMMs are non-discriminative in recognition, i.e. they are not trained as to what to reject in recognition.

There is also a suggestion that a 'weak' model's tokens will be claimed by the 'stronger' model that is characteristically closest to it.

Tests WT7_5 to WT7_11 explore this.

In WT7_5 a test was performed using a single mixture model for `wt1` and double mixture model for the other three tones. Results are in Table 4.5.

In WT7_6 a test was performed using a single mixture model for `wt2` and double mixture model for the other three tones. Results are in Table 4.6.

In WT7_7 a test was performed using a single mixture model for `wt4` and double mixture model for the other three tones. Results are in Table 4.7.

——————————- Overall Results ——————————-
STATS: %Corr=77.53, Acc=77.53 [H=207, D=0, S=60, I=0, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	65	0	0	0	0	100
wt2	0	56	3	0	0	94.9
wt3	0	0	66	0	0	100
wt4	13	1	43	20	0	26.0
Ins	0	0	0	42		

Table 4.7: Results for Test WT7_7

——————————- Overall Results ——————————-
STATS: %Corr=75.28, Acc=73.78 [H=201, D=4, S=62, I=4, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	65	0	0	0	0	100
wt2	6	48	5	0	0	81.4
wt3	1	1	64	0	0	97.0
wt4	21	24	4	24	4	32.9
Ins	3	0	0	1		

Table 4.8: Results for Test WT7_8

In WT7_8 a test was performed using a double mixture model for `wt1` and single mixture model for the other three tones. Results are in Table 4.8.

In WT7_9 a test was performed using a double mixture model for `wt2` and single mixture model for the other three tones. Results are in Table 4.9.

In WT7_10 a test was performed using a double mixture model for `wt3` and single mixture model for the other three tones. Results are in Table 4.10.

In WT7_11 a test was performed using a double mixture model for `wt4` and single mixture model for the other three tones. Results are in Table 4.11.

From these results, summaries of which appear in Tables 4.12 and 4.13, it can be seen that their is a correlation between tones that have similarities in their tone contours: There

——————————- Overall Results ——————————-
STATS: %Corr=62.55, Acc=57.68 [H=167, D=13, S=87, I=13, N=267]
————————————————————————————————————-

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	8	55	0	0	1	12.7
wt2	0	59	0	0	0	100
wt3	1	10	55	0	0	83.3
wt4	9	7	5	45	9	68.2
Ins	0	7	3	55		

Table 4.9: Results for Test WT7_9

——————————- Overall Results ——————————-
STATS: %Corr=63.67, Acc=57.30 [H=170, D=17, S=80, I=17, N=267]
————————————————————————————————————-

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	34	1	6	16	7	59.6
wt2	0	35	21	1	2	61.4
wt3	0	0	65	0	1	100
wt4	2	5	28	36	6	50.7
Ins	1	0	14	1		

Table 4.10: Results for Test WT7_10

——————————————- Overall Results ——————————————-
STATS: %Corr=53.18, Acc=50.56 [H=142, D=7, S=118, I=7, N=267]
——-

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	2	0	0	60	2	3.2
wt2	0	53	1	3	2	93.0
wt3	0	0	10	54	2	15.6
wt4	0	0	0	77	0	100
Ins	0	0	0	7		

Table 4.11: Results for Test WT7_11

when weak	claimed by
wt1	wt2 and wt4
wt2	wt3
wt3	wt4
wt4	wt3

Table 4.12: Summary of Tests WT7_4 to WT7_7

is not much correlation between tones 1 and 3; neither is there much between tones 2 and 4. It is also interesting to note that when a model is 'strong' compared to all others, its correctness rises to 100%.

4.3.2 HHEd

It was attempted to increase the number of mixtures per state to 3 in each model, but HInit's clustering algorithm could not achieve the required number of clusters.

There is an alternative way of increasing the number of mixtures: using HTK's HHEd. HHEd is a HMM editor that allows one to take an existing HMM and modify it in some way, the result being a new model that will provide initial parameter values.

In Test WT7_12, the single mixture models from WT7_2 were taken and a mixture added[12] to each state in each model. These new models were then reestimated using HRest.

The results are shown in Table 4.14.

[12]This was done using HHEd's `MU` command.

when strong	claims
wt1	wt4
wt2	wt1
wt3	wt2 and wt4
wt4	wt1 and wt3

Table 4.13: Summary of Tests WT7_8 to WT7_11

———————————- Overall Results ———————————-
STATS: %Corr=82.77, Acc=81.65 [H=221, D=3, S=43, I=3, N=267]
————————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	64	0	0	0	0	100
wt2	0	57	0	2	0	96.6
wt3	0	7	58	0	1	89.2
wt4	1	31	2	42	1	55.3
Ins	0	2	1	42		

Table 4.14: Results for Test WT7_12

——————————- Overall Results ——————————-
STATS: %Corr=86.14, Acc=86.14 [H=230, D=0, S=37, I=0, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	65	0	0	0	0	100
wt2	1	58	0	0	0	98.3
wt3	0	2	64	0	0	97.0
wt4	12	16	6	43	0	55.8
Ins	0	0	0	42		

Table 4.15: Results for Test WT7_13

——————————- Overall Results ——————————-
STATS: %Corr=86.89, Acc=86.89 [H=232, D=0, S=35, I=0, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	65	0	0	0	0	100
wt2	0	58	1	0	0	98.3
wt3	0	2	64	0	0	97.0
wt4	9	19	4	45	0	58.4
Ins	0	0	0	42		

Table 4.16: Results for Test WT7_14

Although the results are an improvement over those of WT7_2, they are not as good as those of WT7_3. In particular `wt4` seems poorly modelled.

Tests WT7_13 to WT7_15 increased by one the number of mixtures per state for each model to see if results comparable to those of WT7_3 could be achieved. The results are shown in Tables 4.15 to 4.17.

The accuracy peaks in WT7_14 (3 mps per model), and at 86.89% it is still well below that of WT7_3.

Rabiner, Juang, Levinson & Sondhi (1985) found that multimixture CHMMs 'are most sensitive to estimation errors in the location of the means of each mixture density'. If the initial estimate of the mean is not a very good one, the reestimation procedure cannot be expected to yield good parameter estimates.

——————————————- Overall Results ——————————————-
STATS: %Corr=86.14, Acc=86.14 [H=230, D=0, S=37, I=0, N=267]
——-

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	65	0	0	0	0	100
wt2	0	58	1	0	0	98.3
wt3	0	2	64	0	0	97.0
wt4	5	25	4	43	0	55.8
Ins	0	0	0	42		

Table 4.17: Results for Test WT7_15

HHEd's mixture-increasing algorithm (Young 1992, 75-6) works by halving the largest mixture (i.e. halving the weight of the mixture with the largest weight), cloning this, increasing the mean of one by 0.2 standard deviations and decreasing the mean of the other by 0.2 standard deviations.

The above results suggest that when using multiple mixtures it would be best to use HInit to get good initial estimates of the model parameters, rather than using a modified version of an existing model.

On this basis, all the tests used HInit to create initial estimates of multimixture models, rather than using HHEd.

4.4 Summary of WT7 tests

The best results were obtained in Test WT_3, where all models had roughly the same degree of modelling, and whose initial estimates were obtained using HInit.

4.5 The WT26 and WT29 tests

The next set of tests are not as investigative as those using a 7 feature vector (Sections 4.2 and 4.3). They aim to produce results which can be compared with the best of the WT7 tests.

Test WT26_1 used 4 single mixture models; Test WT26_2 used 4 double mixture models; Test WT26_3 used 4 triple mixture models. The results are shown in Tables 4.18 to 4.20.

All the accuracy scores are around the mediocre 70 - 75% mark, which is well below the highest score (94.8%) when only F0-related information is used. Increasing the number

——————————- Overall Results ——————————-
STATS: %Corr=75.28, Acc=70.79 [H=201, D=12, S=54, I=12, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	47	0	0	14	4	77.0
wt2	0	43	3	8	4	79.6
wt3	0	2	56	6	2	87.5
wt4	0	8	13	55	1	72.4
Ins	0	5	4	2		

Table 4.18: Results for Test WT26_1

——————————- Overall Results ——————————-
STATS: %Corr=68.91, Acc=62.92 [H=184, D=16, S=67, I=16, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	35	13	0	12	5	58.3
wt2	0	54	4	0	1	93.1
wt3	0	6	45	12	3	71.4
wt4	1	5	14	50	7	71.4
Ins	2	5	4	2		

Table 4.19: Results for Test WT26_2

——————————— Overall Results ———————————
STATS: %Corr=73.41, Acc=69.66 [H=196, D=9, S=62, I=10, N=267]
————————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	34	21	0	6	4	55.7
wt2	0	58	1	0	0	98.3
wt3	0	5	53	6	2	82.8
wt4	2	13	8	51	3	68.9
Ins	3	4	1	1		

Table 4.20: Results for Test WT26_3

——————————— Overall Results ———————————
STATS: %Corr=76.03, Acc=71.16 [H=203, D=12, S=52, I=13, N=267]
————————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	48	0	0	13	4	78.7
wt2	4	51	2	1	1	87.9
wt3	0	8	50	2	6	83.3
wt4	1	11	10	54	1	71.1
Ins	3	3	3	3		

Table 4.21: Results for Test WT29_1

of mixtures did not make much difference (at least up to 3 mixtures). This time, `wt2` is 'strong' in Tests WT26_2 and Test WT26_3, but claims tokens from a 'weaker' `wt1`.

Test WT29_1 used 4 single mixture models; Test WT29_2 used 4 double mixture models. The results are shown in Tables 4.21 and 4.22. Clustering could not be achieved for 3 mps.

The results compare favourably with those of the WT26 tests, and with WT7_2 in the single mixture case. There is also a more even spread in the correctness scores of the individual models.

——————————————— Overall Results ———————————————
STATS: %Corr=85.39, Acc=84.64 [H=228, D=1, S=38, I=2, N=267]
——

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	56	1	0	8	0	86.2
wt2	1	54	1	3	0	91.5
wt3	0	7	51	7	1	78.5
wt4	1	4	5	67	0	87.0
Ins	1	1	0	42		

Table 4.22: Results for Test WT29_2

Summary of WT26 and WT29 tests

It can be seen that while using only MFCC information, an accuracy score of 75.3%[13] (Test WT26_1) suggests a considerable degree of correlation between tone and syllable, or at least that MFCCs implicitly contain tone contour information.

This was to be expected from the findings of Lin et al. (1993). However, by using explicit F0 information better performance can be achieved.

4.6 Summary of WT tests

When using only 4 models to recognise tone, using the 7-element feature vector (with F0-, and energy-information) yields the best results.

Only using MFCC information performs relatively poorly but does suggest that there is considerable tonal information in MFCCs.

[13]This figure has been adjusted to allow for the error produced by HResults.

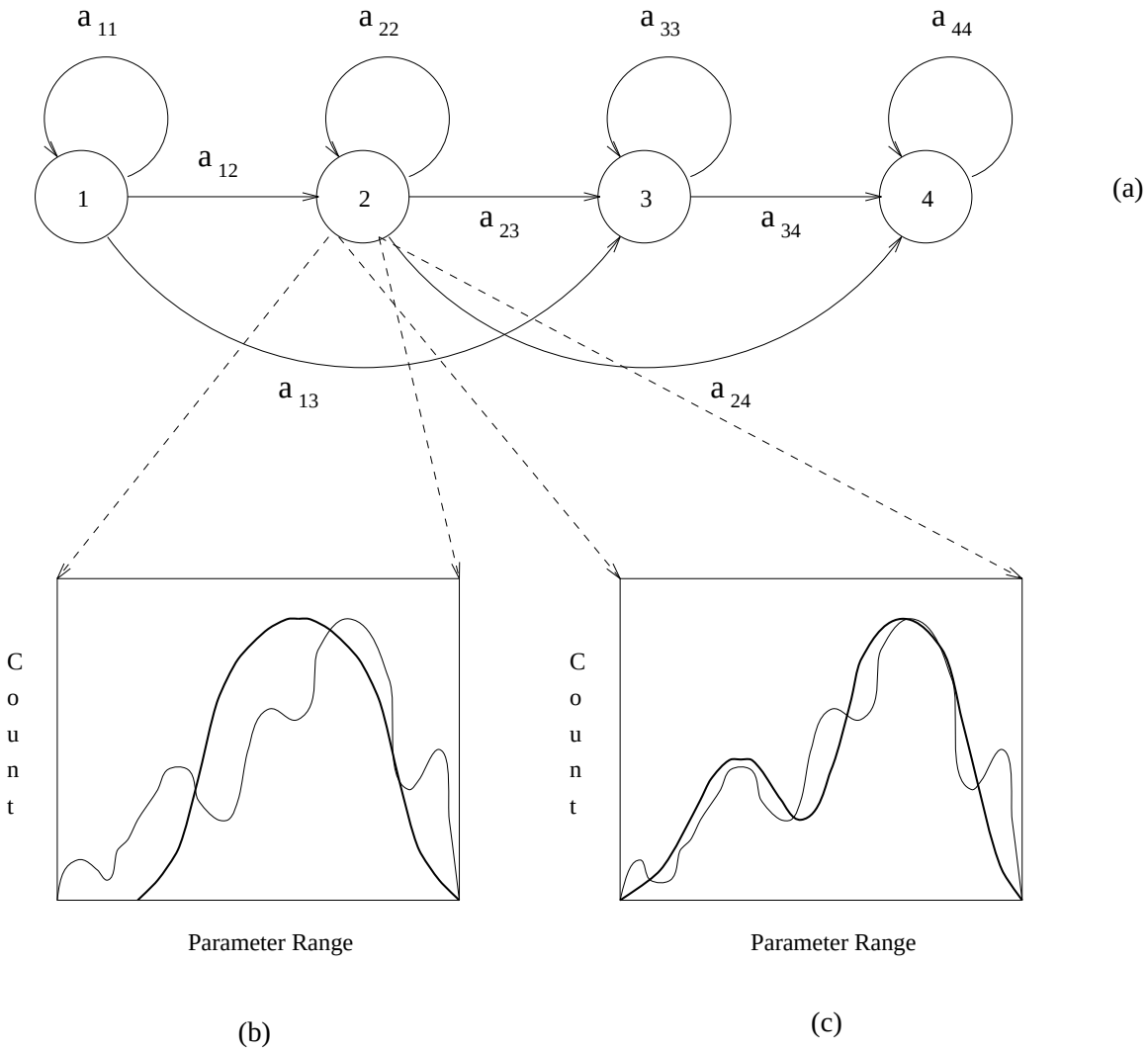

Figure 4.1: (a) 4 state, double transition allowed HMM; (b) Estimated density (thin contour) and single mixture model density (thick contour) for one observation vector component; (c) Same estimated density (thin contour) and double mixture model density (thick contour) for one observation vector component.

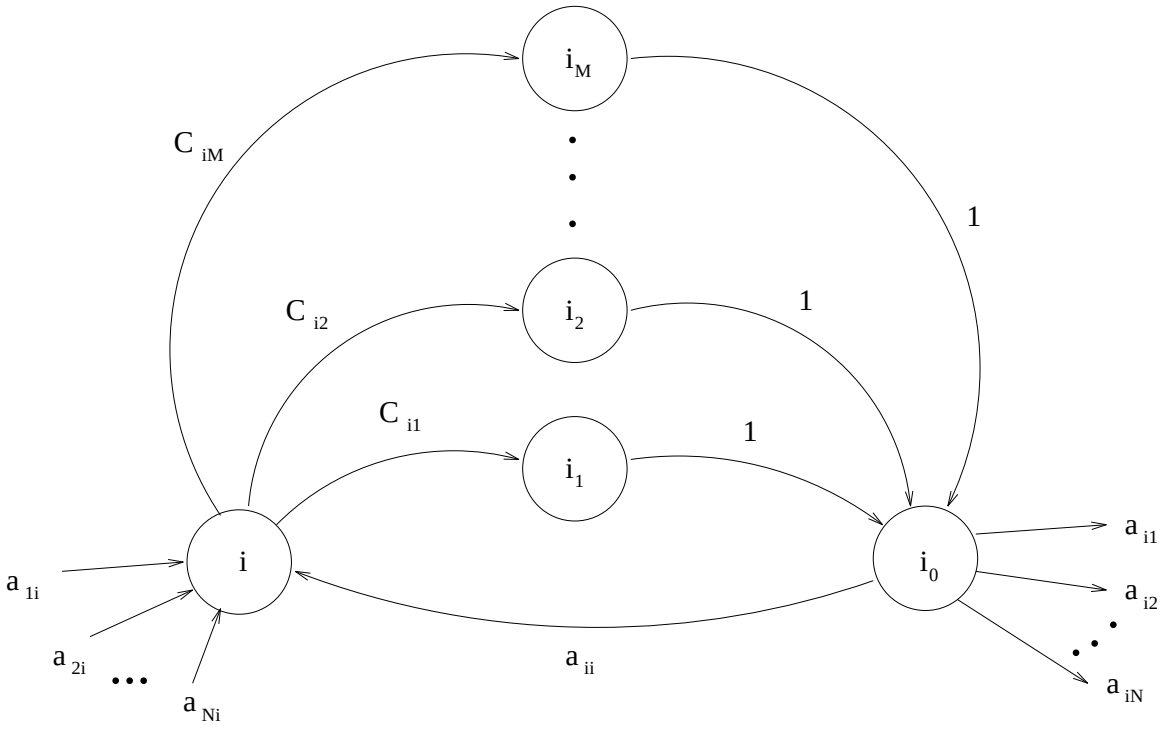

Figure 4.2: Equivalence of a state with a mixture density to a multistate single-density distribution (after Juang et al. (1986)).

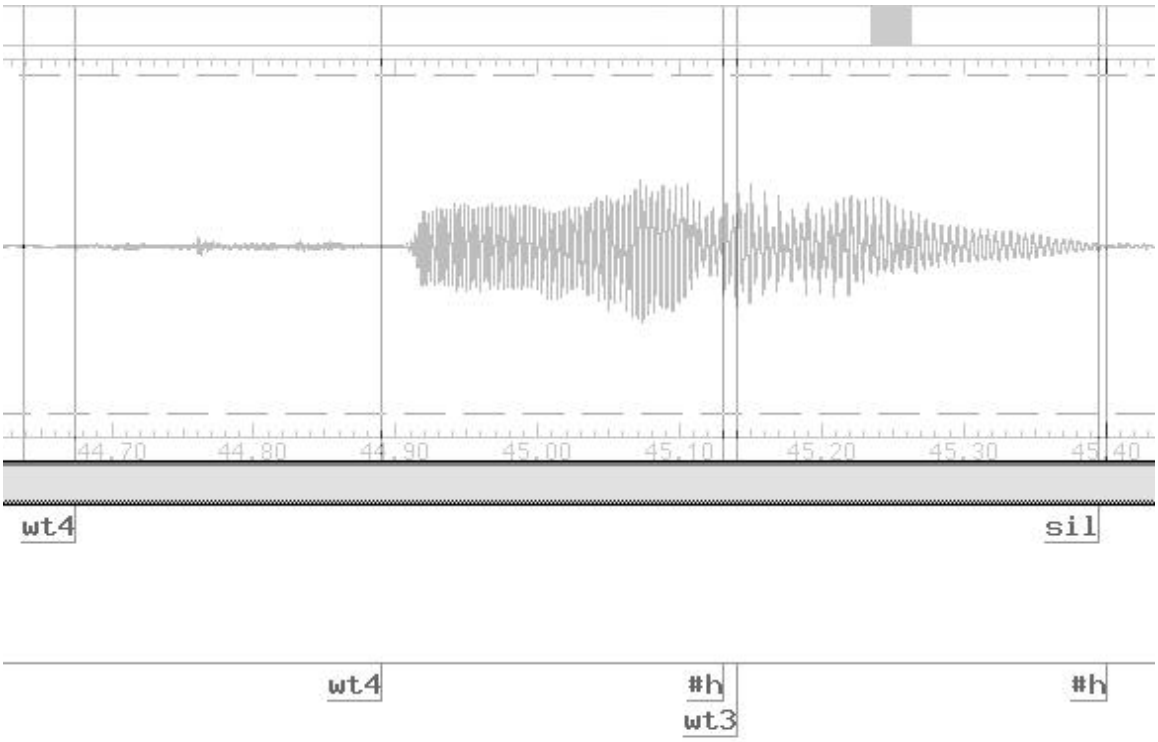

Figure 4.3: Non-speech insertion in word

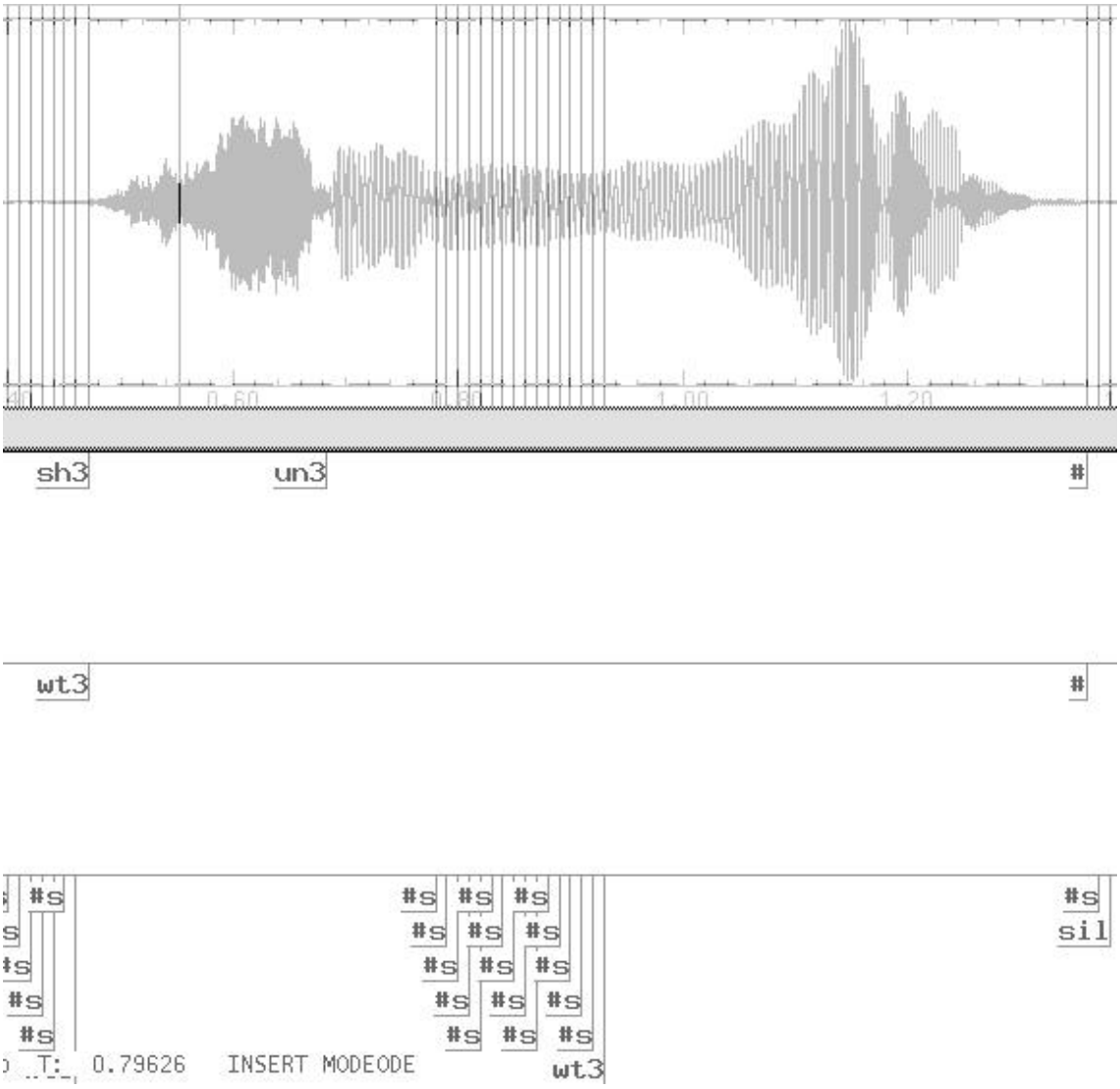

Figure 4.4: 15 non-speech (lipsmack) insertions in tone 3 word

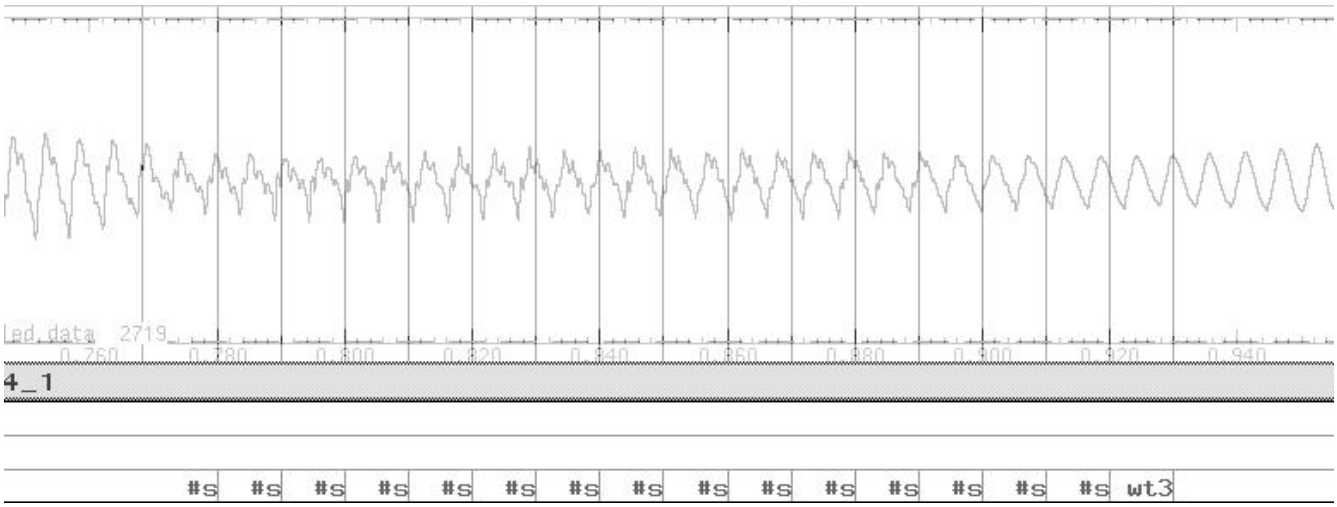

Figure 4.5: 15 lipsmack insertions in tone 3 word (zoom version of previous figure)

Chapter 5

More Recognisers

5.1 The TS and TF tests

The TS (toned segment) TF (toned FINAL) tests took the approach taken by Lin et al. (1993), i.e. to simultaneously recognise tone and syllable.

Because of the limited training data, and to avoid the need to have 268 models, each toned syllable was split into INITIAL and FINAL.

In the TS tests, both segments, INITIAL and FINAL were tone-marked. This reduced the number of models to 104[1] and provided more training data for each model. No label editing was necessary for the TS tests.

In the TF tests, only FINALs were tone-marked. This reduced the number of models even further, to 59[2] and provided more training data for each INITIAL model than was provided in the TS tests. Some label editing was required to replace all the tone-marked INITIALs with untoned equivalents.

If the INITIAL has no influence on the tone, we would expect the tone recognition performance of both the TS and TF tests to be similar. If,however, the INITIAL does exert some influence on the tone, as found by Howie and Ho, we would expect the tone recognition performance of the TS recogniser to be better.

An advantage to using the split-syllable approach is that a more refined grammar can be implemented. For example: a *uu* can only follow a *n, l, x*, or *q*; a *I* can only follow a *s, sh, c*, or *ch*.

No label editing was necessary for the TS tests.

Model topology was an issue that had to be considered.

In the WT series of tests, all models had 4 states, double transition allowed. All models had similar correctness rates when all had the same number of mixtures.

Hon et al. (1994) used three states to model all the context independent sub-syllabic units (INITIALs and FINALs as in this project).

If a fixed number of states is used across all models, then models that have states that

[1] 11 INITIALs, 15 FINALs, 4 tones: $(11 + 15) \times 4 = 104$.

[2] 11 INITIALs, 15 FINALs, 4 tones: $(11 \times 4) + 15 = 59$.

——————————————— Overall Results ———————————————
STATS: %Corr=91.01, Acc=90.26 [H=243, D=2, S=22, I=2, N=267]

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	57	0	0	6	2	90.5
wt2	8	51	0	0	0	86.4
wt3	0	3	63	0	0	95.5
wt4	1	2	2	72	0	93.5
Ins	0	2	0	55		

Table 5.1: Results for Test TS7_1

roughly correspond to to the same number of sounds are expected to perform best (Rabiner & Juang 1993). Picone (1989) in his study of speech recognition suggests that the number of states be allowed to vary across models for better results.

Lin et al. (1993) used 1 state to model INITIALs, 1 state to model the transition between INITIAL and FINAL , and 3 states to model the FINAL. Concatenated, this meant 5 states per model, except for initial-less syllables for which the figure was 4. This method was not an option in this study, unless all the data was relabelled to include a transition state.

The difficulty in deciding the number of states is compounded because we are simultaneously dealing with two entities: the segment *and* the tone.

It was decided to try the following compromise:

All segments were assigned 3 states per model except:

irises: 4 states; an extra state for the non-syllabic *i* .

urns: 5 states; an extra state each for the non-syllabic *u* and the nasal.

All the tests were done using single mixture models; multimixturing was not attempted, as some models already had small numbers of training examples.

The procedure for finding the tone recognition accuracy rates was as follows:

1. Recognise the segments.
2. Using the HLEd label editor, convert to `wt` labels, i.e. words (syllables) marked by tone.
3. Run HResults.

5.1.1 Tone recognition results

The results are displayed in Tables 5.1 to 5.6 with a summary in Table 5.7.

———————————- Overall Results ———————————-
STATS: %Corr=87.64, Acc=86.14 [H=234, D=4, S=29, I=4, N=267]
———————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	57	2	0	4	2	90.5
wt2	2	52	4	0	1	89.7
wt3	0	4	60	2	0	90.9
wt4	0	2	9	65	1	85.5
Ins	0	2	2	55		

Table 5.2: Results for Test TF7_1

———————————- Overall Results ———————————-
STATS: %Corr=89.51, Acc=89.51 [H=239, D=0, S=28, I=0, N=267]
———————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	50	9	0	6	0	76.9
wt2	0	56	2	1	0	94.9
wt3	0	9	57	0	0	86.4
wt4	0	1	0	76	0	98.7
Ins	0	0	0	55		

Table 5.3: Results for Test TS26_1

——————————- Overall Results ——————————-
STATS: %Corr=88.01, Acc=87.64 [H=235, D=1, S=31, I=1, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	49	10	0	6	0	75.4
wt2	0	56	1	1	1	96.6
wt3	0	9	57	0	0	86.4
wt4	0	4	0	73	0	94.8
Ins	0	1	0	55		

Table 5.4: Results for Test TF26_1

——————————- Overall Results ——————————-
STATS: %Corr=92.51, Acc=92.51 [H=247, D=0, S=20, I=0, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	55	4	0	6	0	84.6
wt2	0	56	2	1	0	94.9
wt3	0	5	60	1	0	90.9
wt4	1	0	0	76	0	98.7
Ins	0	0	0	55		

Table 5.5: Results for Test TS29_1

——————————————- Overall Results ——————————————-
STATS: %Corr=89.14, Acc=89.14 [H=238, D=0, S=29, I=0, N=267]
————————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	48	10	0	7	0	73.8
wt2	1	54	2	2	0	91.5
wt3	0	6	60	0	0	90.9
wt4	0	1	0	76	0	98.7
Ins	0	0	0	55		

Table 5.6: Results for Test TF29_1

Test type	TF (toned FINAL)	TS (toned segment)
7_1	87.64	91.01
26_1	88.01	89.51
29_1	89.14	92.51

Table 5.7: % Summary of accuracy scores for tone recognition in the TF and TS tests.

In all 3 cases the marking of the tone of the INITIAL (TS tests) yields better results that when the INITIAL is unmarked for tone. This would seem to confirm that the INITIAL affects the tone and that this could be used to improve recognition.

The best scores are produced by using both F0 *and* MFCC information.

The figures of 88.01% and 89.01% are not as high as as the figure of 94.85% achieved by Lin et al. (1993) who used a 5-state CHMMs to simultaneously recognise tone and syllable (See Section 1.1).

There is a possibility that by using some multimixture models, the figures achieved above could be improved.

5.1.2 Segment recognition results

The results for the recognition of the segments (without merging toned segments together), are shown in Table 5.8.

The figures are not as high as might be expected for two main observed reasons:

1. There were many insertions of vcs at the beginning of initial-less FINALs.
2. The majority of irises and urns were misrecognised as mons.

Test type	TF (toned FINAL)	TS (toned segment)
7_1	25.21	24.79
26_1	60.83	62.50
29_1	63.33	58.96

Table 5.8: % Accuracy scores for segment recognition in the TF and TS tests.

There are at least two possible explanations (which are not mutually exclusive) for these phenomena:

1. The misrecognised segments are poorly modelled relative to other models (See Section 4.3.1).
2. There is a disparity in the numbers of training sequences across models.

5.2 The TSG tests

The TSG (toned segment groups) series of tests employ the principle of the TS tests above, but are only aimed at recognising tone. The segments were placed in 6 groups (as listed in Section 2.4) and were tone-marked. This meant 24 models[3]. It was decided to experiment further with the number of model states.

mon 2 states

iris 4 states, double transition allowed

urn 4 states, double transition allowed

vlf 2 states

asaf 3 states

vc 2 states

The results for the single mixture case are shown in Tables 5.9 to 5.5

Each group above has a different number of elements, i.e. the `mon` group has 7 elements; the `iris` and `urn` groups have 4 each; the `vlf` group has 5; the `asaf` and`vc` groups have 3 each. If the number of mixtures per state in every model is increased in 'in one fell swoop', some models might improve, while at the same time others may be adversely affected. Therefore a PERL program was written to increase the number of mixtures per state in each model, but doing so one at time.

[3]6 groups; 4 tones.

——————————- Overall Results ——————————-
STATS: %Corr=79.78, Acc=76.78 [H=213, D=8, S=46, I=8, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	38	0	0	22	4	63.3
wt2	3	53	0	3	0	89.8
wt3	0	9	54	2	0	83.1
wt4	0	6	1	68	2	90.7
Ins	1	5	0	1		

Table 5.9: Results for Test TSG7_1

——————————- Overall Results ——————————-
STATS: %Corr=90.26, Acc=89.89 [H=241, D=1, S=25, I=1, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	59	2	0	4	0	90.8
wt2	0	53	4	2	0	89.8
wt3	0	1	58	6	1	89.2
wt4	0	2	4	71	0	92.2
Ins	0	0	1	55		

Table 5.10: Results for Test TSG26_1

——————————————— Overall Results ———————————————
STATS: %Corr=85.39, Acc=84.27 [H=228, D=3, S=36, I=3, N=267]
——

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	48	6	0	10	1	75.0
wt2	1	56	0	1	1	96.6
wt3	0	7	57	2	0	86.4
wt4	2	5	2	67	1	88.2
Ins	1	1	0	1		

Table 5.11: Results for Test TSG29_1

5.2.1 Accuracy Maximisation Program

There is no guarantee that increasing the number of mixtures per state will create a better model of the estimated density. Therefore some criterion had to be used in deciding if increasing the number of mixtures per state does actually create a better model of the estimated density. If such a better model was created, it was retained; if not, it was discarded, the older version reinstated, and no more attempts to increase the number of mixtures per state in that model would be made.

The algorithm for deciding if increasing the number of mixtures per state creates a 'better' model of the estimated density is as follows:

```
if    (
      cannot create new model, i.e. clustering cannot be achieved
      )
      then  {
            use older version for next test;
            move on to next model
            }

else  {
      if    (
            accuracy score of segment recognition in current test
            >= highest accuracy score of segment recognition so far
            )
            then  {
                  discard older model;
                  use new version for next test
                  }
```

```
else if (
      (accuracy score of segment recognition in current test
      = highest accuracy score of segment recognition so far)
      and
      (correctness score of segment recognition in current test
      >= highest correctness score of segment recognition so far)
      )
      then  {
            discard older model;
            use new version for next test
            }

else if (
      accuracy score of tone recognition in current test
      >= highest accuracy score of tone recognition so far
      )
      then  {
            discard older model;
            use new version for next test
            }

else if (
      (accuracy score of tone recognition in current test
      = highest accuracy score of tone recognition so far)
      and
      (correctness score of tone recognition in current test
      > highest correctness score of tone recognition so far)
      )
      then  {
            discard older model;
            use new version for next test
            }

else  {
      discard new model;
      use older version for next test
      }
}
```

Sample outputs are shown in Appendix A.

The results for the tests that achieve the highest accuracy are shown in Tables 5.12 to 5.14, and summarised in Table 5.15.

——————————- Overall Results ——————————-
STATS: %Corr=99.63, Acc=99.63 [H=266, D=0, S=1, I=0, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	65	0	0	0	0	100
wt2	0	58	1	0	0	98.3
wt3	0	0	66	0	0	100
wt4	0	0	0	77	0	100
Ins	0	0	0	55		

Table 5.12: Results for Test TSG7_2

——————————- Overall Results ——————————-
STATS: %Corr=93.63, Acc=93.63 [H=250, D=0, S=17, I=0, N=267]
————————————————————————————————

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	62	2	0	1	0	95.4
wt2	0	56	1	2	0	94.9
wt3	0	1	62	3	0	93.9
wt4	1	3	3	70	0	90.9
Ins	0	0	0	55		

Table 5.13: Results for Test TSG26_2

5.2.2 Summary of TSG tests

While using a 29-element feature vector yields high results, the 7-element vector seems slightly better. The 26-element vector is again lagging behind.

The recognition accuracy of the segment groups is still relatively poor with the same error characteristics as were mentioned in Section 5.1.2.

One important point should be noted. The accuracy scores came from using the same data used by the accuracy maximisation program. More objective and reliable results would have been obtained had a fresh set of test data been used.

———————————- Overall Results ———————————-
STATS: %Corr=98.88, Acc=98.88 [H=264, D=0, S=3, I=0, N=267]
———————————————————————————————————-

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	63	1	0	1	0	96.9
wt2	0	59	0	0	0	100
wt3	0	0	65	1	0	98.5
wt4	0	0	0	77	0	100
Ins	0	0	0	55		

Table 5.14: Results for Test TSG29_2

Test type	Tone accuracy	Segment group accuracy
TSG7_2	99.63	57.71
TSG26_2	93.63	66.04
TSG29_2	98.88	58.96

Table 5.15: % Highest Accuracy scores for the TSG tests.

——————————————- Overall Results ——————————————-
STATS: %Corr=99.63, Acc=99.63 [H=266, D=0, S=1, I=0, N=267]
——-

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	64	1	0	0	0	98.5
wt2	0	59	0	0	0	100
wt3	0	0	66	0	0	100
wt4	0	0	0	77	0	100
Ins	0	0	0	55		

Table 5.16: Results for Test TGF7_1

5.3 The TGF tests

The TGF (toned grouped final) series of tests employ the principle of the TSG tests above, and are again only aimed at recognising tone. The INITIAL segments were placed in 3 groups but were not tone-marked. The FINALs were all grouped together and were tone-marked. This meant 7 models as listed below. The makeup of the models that attain the maximum accuracy - 99.63% - using the accuracy maximisation program (See Section 5.2.1) are shown below. It was decided to use the same number of model states for the INITIALs as was used in the TSG tests above. Initially it was decided to use 3 states for each FINAL model, but FINAL1 and FINAL2 would not initialise with 3 states, but would initialise using 2 states.

FINAL1 2 states, 4 mixtures per state.

FINAL2 2 states, 3 mixtures per state.

FINAL3 3 states, 3 mixtures per state.

FINAL4 3 states, 3 mixtures per state.

vlf 2 states, 3 mixtures per state.

asaf 3 states, 3 mixtures per state.

vc 2 states, 3 mixtures per state.

The best results for TGF7 are shown in Table 5.16.

No TGF26 tests were carried out as the accuracy scores for tone recognition using only MFCC information had in the other tests been low relative to the scores achieved using F0

——————————————- Overall Results ——————————————-
STATS: %Corr=97.75, Acc=97.75 [H=261, D=0, S=6, I=0, N=267]
——-

Confusion Matrix						
	wt1	wt2	wt3	wt4	Del	%c
wt1	61	4	0	0	0	93.8
wt2	0	58	1	0	0	98.3
wt3	0	0	66	0	0	100
wt4	0	0	1	76	0	98.7
Ins	0	0	0	55		

Table 5.17: Results for Test TGF29_1

information (7-element feature vector), and F0 and MFCC information (29-element feature vector).

For the TGF29 tests, the makeup of the models that attain the maximum accuracy - 97.75% - using the accuracy maximisation program are shown below.

FINAL1 2 states, 5 mixtures per state.

FINAL2 3 states, 3 mixtures per state.

FINAL3 3 states, 5 mixtures per state.

FINAL4 3 states, 4 mixtures per state.

vlf 2 states, 3 mixtures per state.

asaf 3 states, 4 mixtures per state.

vc 2 states, 2 mixtures per state.

The best results for TGF29 are shown in Table 5.17.

5.3.1 Summary of TGF tests

It is possible to achieve high tonal recognition without using as many models as were used in the TSG tests. On the other hand, more mixtures per state per model were required.

It would seem that tone-marking the FINALSs is sufficient in order to obtain a very high degree of accuracy.

Chapter 6

Conclusion

6.1 Main Points

1. While there is evidence to suggest there is a high degree of correlation between tone and syllable as Lin et al. (1993) argued, there is no evidence to suggest that an integrated tone-syllable approach is better than a tone-independent-of-syllable approach for tone recognition. In fact the results point towards the latter being slightly preferable.

2. The results do confirm two widely accepted things:

   (a) Explicit F0 information is best for tone recognition.

   (b) HMMs are non-discriminative.

3. The results contradict the findings of the Golden Mandarin I and II designers who found that a single mixture Gaussian density was adequate in modelling tone. They said that mixture densities were found to produce slightly lower or equal recognition rates.

4. While grouping segments together according to tone and phonetic makeup (i.e. using the findings of Howie (1974) and Ho (1976)) does achieves a very high degree of accuracy, comparable results can be achieved by simply tone-marking the FINALSs and using multimixture models.

5. It is possible to use only two states to model tones 1 and 2.

6.2 Future Work

In the short term, the icing on the cake would be to see if a fresh set of test data produced accuracy scores comparable to those yielded by the TGF7 and TSG7 tests (99.63%).

In the longer term, there are two main directions this work could go:

1. To conduct similar tests on tone recognition for the following:

    - polysyllables and/or continuous speech;
    - the 5th tone;
    - speaker independence and adaption.

2. To conduct extensive tests on syllable recognition to see how an integrated tone-syllable approach and a syllable-independent-of-tone approach compare.

Bibliography

Bagshaw, P. C., Miller, S. M. & Jack, M. A. (1993), 'Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching', *Proceedings of EUROSPEECH* pp. 1003–1006.

Chan, C. & Ng, K. W. (1982), Intonation recognition of discrete utterances in Mandarin, Technical Report TR-A8-82, Center of Computer Studies Appl., University of Hong Kong, Hong Kong.

Chao, Y. R. (1948), *Mandarin Primer*, Harvard University Press, Cambridge.

Chao, Y. R. (1968), *A Grammar of Spoken Chinese*, University of California Press, Berkeley.

Cheng, C.-C. & Sherwood, B. (1992), 'Technical aspects of computer–assisted instruction in chinese', *Tsing Hua Journal of Chinese Studies* **New Series XIV**(1 and 2), 35–49.

Cheng, P.-C., Sun, S.-W. & Chen, S.-H. (1990), 'Mandarin tone recognition by multi-layer perceptron', *IEEE 1990 International Conference on Acoustics, Speech and Signal Processing* pp. 517–520.

Deller, Jr., J. R., Proakis, J. G. & Hansen, J. H. (1993), *Discrete-Time Processing of Speech Signals*, Macmillan, New York.

Dow, F. D. M. (1972), *An Outline of Mandarin Phonetics*, Australian National University Press, Canberra.

Fu, S. W. K., Lee, C. H. & Clubb, O. L. (1996), 'A survey on Chinese speech recognition', *Chinese and Oriental Languages Information Processing Society Journal*.

Gao, Y., Hon, H.-W., Lin, Z., Loudon, G., Yogananthan, S. & Yuan, B. (1995), 'Tangerine: A large vocabulary Mandarin dictation system', *IEEE 1995 International Conference on Acoustics, Speech and Signal Processing* pp. 77–80.

Garding, E. (1987), 'Speech act and tonal pattern in standard Chinese: constancy and variation', *Phonetica* **44**, 13–29.

Han, M. S. & Kim, K.-O. (1974), 'Phonetic variation of Vietnamese tones in disyllabic utterances', *Journal of Phonetics* **2**, 223–232.

Hieronymous, J., Alexander, M., Bennett, C., Cohen, I., Davies, D., Dalby, J., Laver, J., Barry, W., Fourcin, A. & Wells, J. (1990), *Proposed Speech Segmentation Criteria for the SCRIBE Project*, preliminary draft edn, Centre for Speech Technology Research, University of Edinburgh and Department of Phonetics and Linguistics, University College London. Rev. 2.

Ho, A. T. (1976), 'The acoustic variation of Mandarin tones', *Phonetica* **33**, 353–367.

Hon, H.-W., Yuan, B., Chow, Y.-L., Narayan, S. & Lee, K.-F. (1994), 'Towards large vocabulary Mandarin chinese speech recognition', *IEEE 1994 International Conference on Acoustics, Speech and Signal Processing* pp. 545–548.

Howie, J. M. (1974), 'On the domain of tone in Mandarin', *Phonetica* **30**, 129–148.

ICC (1988), *Proc. 1986, 1987, and 1988 International Conference on Computer Processing of Chinese and Oriental Languages*.

Isard, A. & Thompson, H. (1996), Transcription conventions, Used by Human Communications Research Centre, University of Edinburgh for labelling the Map Task Corpus.

Juang, B.-H., Levinson, S. E. & Sondhi, M. M. (1986), 'Maximum likelihood estimate for multivariate mixture observations of Markov chains', *IEEE Transactions on Information Theory* **IT-32**(2), 307–309.

Kratochvil, P. (1968), *The Chinese Language Today*, Hutchinson University Library, London.

Ladd, D. R. (1996), *Intonational Phonology*, Cambridge University Press. Forthcoming.

Lee, L.-S., Tseng, C.-Y., Chen, K.-J., Hung, I.-J., Lee, M.-Y., , Chien, L.-F., Lee, Y.-M., Lyu, R.-Y., Wang, H.-m., Wu, Y.-C., Lin, T.-S., Gu, H.-Y., Nee, C.-p., Liao, C.-Y., Yang, Y.-J., Chang, Y.-C. & Yang, R.-C. (1993), 'Golden Mandarin (II) - an improved single-chip real-time Mandarin dictation machine for Chinese language with very large vocabulary', *IEEE 1993 International Conference on Acoustics, Speech and Signal Processing* pp. II503–506.

Lee, L.-S., Tseng, C.-Y., Gu, H.-Y., Liu, F.-H., Chang, C.-H., Hsieh, S.-H. & Chen, C.-h. (1990), 'A real-time Mandarin dictation machine for Chinese language with unlimited texts and very large vocabulary', *IEEE 1990 International Conference on Acoustics, Speech and Signal Processing*.

Lee, L.-S., Tseng, C.-Y., Gu, H.-Y., Liu, F.-H., Chang, C.-H., Lin, Y.-H., Lee, Y.-M., Tu, S.-L., Hsieh, S.-H. & Chen, C.-h. (1993), 'Golden Mandarin (I) - real-time Mandarin speech dictation machine for Chinese language with very large vocabulary', *IEEE Transactions on Speech and Audio Processing* **1**(2), 158–179.

Lin, C. H., Lee, L. S. & Ting, P. Y. (1993), 'A new framework for recognition of Mandarin syllables with tones using sub-syllabic units', *IEEE 1993 International Conference on Acoustics, Speech and Signal Processing* pp. 227–230.

Lyu, R.-Y., Chien, L.-F., Hwang, S.-H., Hsieh, H.-Y., Yang, R.-C., Bai, B.-R., Weng, J.-C., Yang, Y.-J., Lin, S.-W., Chen, K.-J., Tseng, C.-Y. & Lee, L.-S. (1995), 'Golden Mandarin (III) - a user-adapted prosodic-segment-based Mandarin dictation machine for Chinese language with very large vocabulary', *IEEE 1995 International Conference on Acoustics, Speech and Signal Processing* pp. 57–60.

Medan, Y., Yair, E. & Chazan, D. (1991), 'Super resolution pitch determination of speech signals', *IEEE Transactions on Signal Processing* **39**, 40–48.

Owens, F. J. (1993), *Signal Processing of Speech*, Macmillan Press Ltd, Hampshire, UK.

Picone, J. (1989), 'On modelling duration in context in speech recognition', *IEEE 1989 International Conference on Acoustics, Speech and Signal Processing* **1**, 421–424.

Rabiner, L. R. (1989), 'A tutorial on hidden Markov models and selected applications in speech recognition', *IEEE Transactions on Acoustics, Speech and Signal Processing* pp. 267–296.

Rabiner, L. R. & Juang, B.-H. (1993), *Fundamentals of Speech Recognition*, Prentice-Hall signal processing series, Englewood Cliffs, N.J. : PTR Prentice Hall.

Rabiner, L. R. & Schafer, R. W. (1978), *Digital Processing of Speech Signals*, Prentice-Hall, New Jersey, USA.

Rabiner, L. R., Juang, B.-H., Levinson, S. E. & Sondhi, M. M. (1985), 'Some properties of continuous hidden Markov model representations', *AT&T Technical Journal* **64**(6), 1251–1270.

Shen, X.-n. S. (1990), *The Prosody of Mandarin Chinese*, Vol. 118 of *University of California publications in linguistics*, University of California Press.

Shih, C.-L. (1988), 'Tone and intonation in Mandarin', *Working Papers of the Cornell Phonetics Laboratory* **3**, 83–109.

Wang, W. S.-Y. (1967), 'Phonological features of tone', *International Journal of American Linguistics* **33**, 93–105.

Yang, W.-j., Lee, J.-c., Chang, Y.-c. & Wang, H.-c. (1988), 'Hidden Markov model for Mandarin lexical tone recognition', *IEEE Transactions on Acoustics, Speech and Signal Processing* **36**(7), 988–992.

Young, S. J. (1992), *HTK: Hidden Markov Model Toolkit V1.4 Reference Manual*, Cambridge University Engineering Department Speech Group.

Appendix A

Sample outputs from accuracy maximisation program

A.1 TSG7 tests

A.1.1 TSG7 Segment Recognition

Test	Model	Sts	M/S	%Corr	%Acc	H	D	S	I	N
1	-	-	-	40.42	27.29	194	38	248	63	480
2	vlf1	2	2	39.79	26.46	191	38	251	64	480
3	asaf1	3	2	40.62	28.12	195	34	251	60	480
4	vc1	2	2	42.29	30.00	203	28	249	59	480
5	mon1	2	2	42.92	32.50	206	27	247	50	480
6	iris1	4	2	42.29	31.46	203	29	248	52	480
7	urn1	4	2	43.12	33.96	207	27	246	44	480
8	vlf2	2	2	44.17	35.62	212	28	240	41	480
9	asaf2	3	2	43.33	34.79	208	28	244	41	480
10	vc2	2	2	46.46	34.79	223	33	224	56	480
11	mon2	2	2	47.92	35.62	230	35	215	59	480
12	iris2	4	2	47.92	35.42	230	35	215	60	480
13	urn2	4	2	47.50	36.88	228	26	226	51	480
14	vlf3	2	2	47.50	36.88	228	26	226	51	480
15	asaf3	2	2	47.50	36.88	228	26	226	51	480
16	vc3	2	2	47.08	36.46	226	25	229	51	480
17	mon3	2	2	52.08	41.04	250	26	204	53	480
18	iris3	4	2	49.79	38.33	239	32	209	55	480
19	urn3	4	2	52.92	41.67	254	28	198	54	480
20	vlf4	2	2	52.92	41.67	254	28	198	54	480
21	asaf4	3	2	52.71	41.67	253	27	200	53	480

22	vc4	2	2	53.33	41.88	256	22	202	55	480
23	mon4	2	2	55.62	44.38	267	18	195	54	480
24	iris4	4	2	56.04	44.79	269	18	193	54	480
25	urn4	4	2	56.04	44.79	269	18	193	54	480
26	vlf1	2	3	56.04	44.79	269	18	193	54	480
27	asaf1	3	3	55.83	45.21	268	15	197	51	480
28	vc1	2	3	56.04	45.00	269	11	200	53	480
29	mon1	2	3	55.83	43.75	268	9	203	58	480
30	vlf2	2	3	55.62	43.54	267	9	204	58	480
31	vc2	2	3	55.83	43.33	268	11	201	60	480
32	mon2	2	3	55.42	44.38	266	16	198	53	480
33	urn2	4	3	52.92	42.08	254	23	203	52	480
34	vlf3	2	3	56.04	44.58	269	18	193	55	480
35	asaf3	2	3	55.83	44.38	268	18	194	55	480
36	mon3	2	3	57.29	48.12	275	20	185	44	480
37	urn3	4	3	57.29	48.12	275	20	185	44	480
38	vlf4	2	3	57.29	48.54	275	18	187	42	480
39	asaf4	3	3	57.71	49.58	277	15	188	39	480
40	vc4	2	3	57.71	49.17	277	15	188	41	480
41	mon4	2	3	57.92	49.58	278	15	187	40	480
42	asaf1	3	4	56.88	48.54	273	15	192	40	480
43	vc1	2	4	57.08	48.75	274	14	192	40	480
44	mon1	2	4	56.88	48.54	273	15	192	40	480
45	vlf2	2	4	56.67	48.33	272	15	193	40	480
46	vc2	2	4	56.67	48.33	272	14	194	40	480
47	mon2	2	4	56.25	47.29	270	9	201	43	480
48	vlf3	2	4	56.25	47.08	270	9	201	44	480
49	asaf3	2	4	56.46	47.29	271	9	200	44	480
50	mon3	2	4	57.08	46.67	274	2	204	50	480

A.1.2 TSG7 Tone Recognition

Test	Model	Sts	M/S	%Corr	%Acc	H	D	S	I	N
1	-	-	-	79.78	76.78	213	8	46	8	267
2	vlf1	2	2	80.15	77.53	214	7	46	7	267
3	asaf1	3	2	80.52	77.90	215	7	45	7	267
4	vc1	2	2	84.27	81.65	225	7	35	7	267
5	mon1	2	2	80.52	78.28	215	6	46	6	267
6	iris1	4	2	80.15	77.90	214	6	47	6	267
7	urn1	4	2	79.78	77.53	213	6	48	6	267
8	vlf2	2	2	80.52	78.65	215	5	47	5	267
9	asaf2	3	2	79.40	77.90	212	4	51	4	267
10	vc2	2	2	81.27	80.15	217	3	47	3	267
11	mon2	2	2	81.65	80.90	218	2	47	2	267
12	iris2	4	2	81.27	80.90	217	1	49	1	267
13	urn2	4	2	82.40	82.02	220	1	46	1	267
14	vlf3	2	2	82.40	82.02	220	1	46	1	267
15	asaf3	2	2	82.40	82.02	220	1	46	1	267
16	vc3	2	2	83.52	82.77	223	2	42	2	267
17	mon3	2	2	87.64	87.64	234	0	33	0	267
18	iris3	4	2	85.02	85.02	227	0	40	0	267
19	urn3	4	2	87.64	87.64	234	0	33	0	267
20	vlf4	2	2	87.64	87.64	234	0	33	0	267
21	asaf4	3	2	88.76	88.76	237	0	30	0	267
22	vc4	2	2	89.51	89.51	239	0	28	0	267
23	mon4	2	2	93.26	93.26	249	0	18	0	267
24	iris4	4	2	93.63	93.63	250	0	17	0	267
25	urn4	4	2	93.63	93.63	250	0	17	0	267
26	vlf1	2	3	93.63	93.63	250	0	17	0	267
27	asaf1	3	3	93.63	93.63	250	0	17	0	267
28	vc1	2	3	94.38	94.38	252	0	15	0	267
29	mon1	2	3	94.38	94.38	252	0	15	0	267
30	vlf2	2	3	94.38	94.38	252	0	15	0	267
31	vc2	2	3	94.01	94.01	251	0	16	0	267
32	mon2	2	3	94.01	94.01	251	0	16	0	267
33	urn2	4	3	89.14	89.14	238	0	29	0	267
34	vlf3	2	3	94.01	94.01	251	0	16	0	267
35	asaf3	2	3	94.01	94.01	251	0	16	0	267
36	mon3	2	3	97.75	97.75	261	0	6	0	267
37	urn3	4	3	97.75	97.75	261	0	6	0	267
38	vlf4	2	3	97.75	97.75	261	0	6	0	267

39	asaf4	3	3	98.13	98.13	262	0	5	0	267
40	vc4	2	3	98.50	98.50	263	0	4	0	267
41	mon4	2	3	98.88	98.88	264	0	3	0	267
42	asaf1	3	4	98.88	98.88	264	0	3	0	267
43	vc1	2	4	98.88	98.88	264	0	3	0	267
44	mon1	2	4	98.88	98.88	264	0	3	0	267
45	vlf2	2	4	98.88	98.88	264	0	3	0	267
46	vc2	2	4	98.88	98.88	264	0	3	0	267
47	mon2	2	4	99.25	99.25	265	0	2	0	267
48	vlf3	2	4	99.25	99.25	265	0	2	0	267
49	asaf3	2	4	99.25	99.25	265	0	2	0	267
50	mon3	2	4	99.63	99.63	266	0	1	0	267