MEMORY-BASED TONE RECOGNITION OF CANTONESE SYLLABLES

by

Michael Emonts

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Arts

Department of Linguistics

Brigham Young University

December 2002

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Michael Emonts

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Deryle W. Lonsdale, Chair

Date	C. Ray Graham

Date	Matthew B. Christensen

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Michael Emonts in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

___________________________	______________________________________
Date	Deryle W. Lonsdale Chair, Graduate Committee

Accepted for the Department

____

Lynn E. Henrichsen
Chair, Department of Linguistics and English Language

Accepted for the College

____

Van C. Gessel
Dean, College of Humanities

ABSTRACT

MEMORY-BASED TONE RECOGNITION OF CANTONESE SYLLABLES

Michael Emonts

Department of Linguistics

Master of Arts

Speech recognition has only recently been applied to Cantonese. Considerable effort, however, has been spent in recognizing Mandarin, the standard dialect of Chinese. Prior to this thesis, the only published work on monosyllabic Cantonese tone recognition is from Tan Lee et al. (1993,1995). This thesis is the first of its kind in that it explores memory-based learning as a viable approach for Cantonese tone recognition.

The memory-based learning algorithm employed in this thesis outperforms the highly respected and widely used neural network approach. Various numbers of tones and features are modeled to find the best method for feature selection and extraction. To further optimize this approach, experiments are performed to isolate the best feature weighting method, best class voting weights method, and the best number of *k*-values to implement. A detailed error analysis is also reported.

This thesis will prove valuable as a future reference for memory-based learning in application to more complex tasks such as continuous speech tone recognition.

ACKNOWLEDGEMENTS

First and foremost, I would like to acknowledge my loving wife Anita, without whom this work would not be possible. I'm sure many days she felt like a single parent, and in many ways, her sacrifices far surpassed my own. My sincerest apologies go out to Jordan, my one-year-old daughter, as many days she never got to see her 'daddy'.

I would also like to acknowledge the valuable help from Xavier Menéndez-Pidal and Gustavo Hernández-Ábrego, two coworkers who periodically aided me in understanding difficult speech recognition concepts. Many thanks also go out to Sony Electronics for allowing me to use their premises to work on my thesis, and for the frequent use of their vending machines, from which I hope I will never need to eat again.

Recognition must also go out to my thesis committee: Deryle Lonsdale, Ray Graham, and Matt Christensen. They were not only instrumental in the production of this thesis, but also in shaping me as a linguist, having each taught me several classes in preparation for this work. I consider each member of my committee a dear friend and exemplar in my life, both academically and non-academically.

Lastly, I would like to acknowledge my parents, who despite not finishing high school themselves, were able to instill in me a deep love for education. Although I am sure they will never read this thesis, I admit I was often spurred on with a motivation to 'do my parents proud'.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

INTRODUCTION

An indispensable component of any Chinese speech recognizer is a tone recognizer. This thesis is the first of its kind in that it explores memory-based learning as a viable approach for Cantonese tone recognition.

Speech recognition has only recently been applied to Cantonese. Considerable effort, however, has been spent in recognizing Mandarin, the standard dialect of Chinese. Prior to this thesis, the only published work on monosyllabic Cantonese tone recognition is from Tan Lee et al. (1995) at the Chinese University of Hong Kong.

Using neural network tone recognition, Lee et al. reported 89.0% accuracy in their speaker-dependent system. The memory-based learning algorithm employed in this thesis is shown to obtain 90.9% in a speaker-dependent system (81.8% in a speaker-independent system), thus outperforming the highly respected and widely used neural network approach.

Various numbers of tones and features are modeled to find the best method for feature selection and extraction. To further optimize this approach, experiments are performed to isolate the best feature weighting method, best class voting weights method, and the best number of *k*-values to implement. A detailed error analysis is also reported.

It is hoped that this thesis will prove valuable as a future reference for memory-based learning in application to more complex tasks, such as continuous speech recognition.

This thesis comprises 4 chapters. Chapter 1 presents a brief introduction of Cantonese phonology, emphasizing areas relevant to tone recognition as well as a review of published literature regarding tone recognition. Chapter 2 summarizes the methodology of the approach used in this thesis. Results and conclusions are then presented in Chapters 3 and 4 respectively.

1 Review of Literature

In order to better understand the issues regarding Cantonese tone recognition, a review of literature is presented in this chapter. Following a brief introduction of the Cantonese language, aspects of Cantonese phonology relevant to tone recognition are presented. A detailed discussion of Cantonese tones follows, with the chapter concluding with a brief overview of published research in tone recognition.

1.1 Cantonese Language

Cantonese is one of many dialects of Chinese. It should be noted that the term ‘dialect’ might be misleading, as dialects of Chinese are not necessarily mutually intelligible. Most dialects of Chinese do, however, share a common writing system. This fact, along with cultural and political motivations, justifies its classification as a dialect of Chinese. There exists no standardized written form for Cantonese. Cantonese-speakers use the standard Chinese writing style which is very similar to Mandarin in terms of vocabulary and grammar.

Cantonese is primarily spoken in the southern Chinese provinces of Guangdong and Gwongxi as well as in the Chinese territories of Hong Kong and Macau. Emigration from these areas has lead to scattered Cantonese-speaking communities throughout the world. Varieties of Cantonese can be heard in Singapore, Malaysia, as well as in various cities in Australia, Europe, and North America.

Worldwide, there are over 40 million native speakers of Cantonese, or approximately 4.0% of the total population of China. Cantonese pales in comparison to

Mandarin, which is spoken by over 700 million speakers, or 71.5% of the total population of China. In fact, Cantonese only ranks 4th among Chinese dialects as there are more speakers of Wu (7.5%) and Min (5.6%) than Cantonese (Bauer and Benedict 1997).

Linguistically, Cantonese receives more attention than other Chinese dialects such as Wu and Min. Cantonese-speaking areas in China, as well as Hong Kong and Macau have prospered economically. Until recently, the vast majority of overseas Chinese were Cantonese-speakers for this reason. Many areas of China also consider Cantonese as a prestigious dialect of Chinese. For this same reason, many now consider Cantonese as spoken in Hong Kong to be the Cantonese standard.

Cantonese has a complicated tone system (see Chapter 1.3.1). To illustrate the importance of tones, consider the antonyms *maai* ‘to buy’ and *maai* ‘to sell’. The first word is pronounced with a low-rising tone; the second with a low-level tone. Even a slight change in intonation may result in a severe communication problem. The importance of tones can be further illustrated with the fact that 98.9% of Chinese have a surname that differs from another surname by tone only (Zhang et al. 2002).

1.2 Cantonese Syllables

1.2.1 Syllable Structure

Syllable structure for Chinese, and particularly in Cantonese, is extremely simple. Cantonese syllables only occur in four variations: V, CV, VC, or CVC. This can be collectively represented as (C)V(C). The initial consonant may be any valid consonant in Cantonese, or none at all. V is a mandatory vowel which may also be a diphthong. The optional final consonant may only be a nasal (i.e. /m/, /n/, or /ng/) or a voiceless unaspirated stop (i.e. /p/, /t/, /k/). None of the consonants may be a consonant cluster.

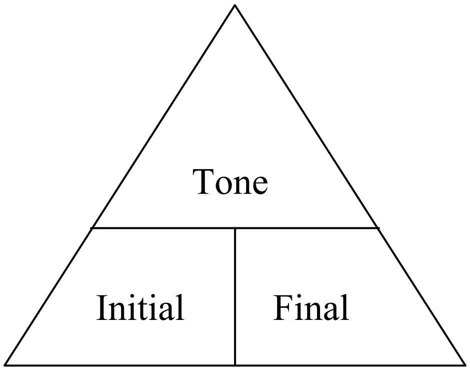


Figure 1-1 A Cantonese Syllable's 3 Parts: Initial, Final, and Tone

Chinese linguists have traditionally viewed the Chinese syllable structure to consist of an initial part, a final part, and an overlying tone (Figure 1-1). The initial part consists of either a voiced or an unvoiced consonant in Cantonese; it may also be null. The final, or rime, is comprised of a mandatory vowel or diphthong followed by an optional coda. The tone is an intonational feature extending over the entire syllable.

Cantonese is one of several languages in which tonal contour carries lexical meaning. In other words, the tone associated with a syllable carries an essential element of the syllable, without which comprehension is very difficult. A Cantonese syllable spoken in isolation with an incorrect tone is just as difficult to comprehend as if one of the syllable's phonemes were replaced with another. Figure 1-2 further illustrates this by providing 9 different Cantonese words that differ by tone only. The romanization system as well as an introduction to each of the tones will be introduced in the next section.

Romanization	Character	Meaning
si1	絲	‘poem’
si2	使	‘history’
si3	試	‘to test’
si4	時	‘time’
si5	市	‘market’
si6	事	‘thing’
sik7	色	‘color’
sip8	攝	‘to absorb’
sik9	食	‘to eat’

Figure 1-2 Syllable *si* Contrasted in 9 Different Tones

1.2.2 Unaspirated Stops

An interesting feature in Cantonese is the potential for unaspirated stops at the end of syllables. Unaspirated stops are a vestige from Middle Chinese still preserved in modern Cantonese and provide evidence supporting the antiquity of the Cantonese dialect. Figure 1-3 shows an example of each of the three unaspirated stops in Cantonese.

Romanization	Character	Meaning
sat1	失	‘to lose’
sak1	塞	‘to block’
sap1	濕	‘wet’

Figure 1-3 Examples of Unaspirated Stops in Cantonese

Each of the three unaspirated stops (either /p/, /t/, or /k/) is pronounced like the regular consonant except that it is unreleased. The [t] in *sat* is pronounced by touching the tongue to the alveolar ridge behind the teeth, but then no air is released. Similarly, the [p] in *sap* is pronounced by bringing both lips together, but then are not reopened, thus preventing the release of air.

Distinguishing between the three can prove very difficult for learners of Cantonese as all three tend to sound like a glottal stop. Nevertheless, it is vitally important for Cantonese speakers to do so in order to distinguish between the minimal pairs presented above in Figure 1-3.

As will be shown later, unaspirated stops play a very important part in Cantonese tone recognition as they are tied to three of the tones in Cantonese.

1.2.3 Vowel Length

Another interesting aspect of Cantonese is that vowel length is significant. The word *gai* 'chicken', for example, is pronounced with a very short diphthong. The word *gaai* 'street' contains a longer diphthong. Each word differs in vowel length and of course in lexical meaning.

In Cantonese, there are 13 vowels (7 long, 6 short) and 10 diphthongs (4 long, 6 short). Figure 1-4 shows the Cantonese vowel system with long vowels in bold. Vowel pairs (e.g. /ɪ/ and /**i**/) are allophones of each other, conditioned by the syllable coda. It is clear from Figure 1-4 that Cantonese allophones differ by not only vowel length but also vowel quality. Interestingly, each of the short vowel allophones is more neutralized than its longer equivalent. The difference in vowel quality is clearly audible in monophthongs such as the distinction between *sam1* 'heart' and *saam1* 'three'.

	Front		Central	Back
	Unrounded	Rounded		
High	**i**	**yu**		**u**
Mid-high	i			u
Mid	e	eu		o
Mid-low	**e**	**eu**	a	**o**
Low			**aa**	

Figure 1-4 Cantonese Vowels (Based on Bauer and Benedict 1997)

Figure 1-5 shows the Cantonese diphthongs. Again, the long vowels are in bold face. With diphthongs, a difference in vowel quality is arguably undetectable as, for example, in the *gai*/*gaai* distinction mentioned earlier. Unlike Mandarin, there are no triphthongs in Cantonese.

Base Vowel	Diphthong	
	+ i	+ u
i		iu
e	ei	
eu		
eu	eui	
a	ai	au
aa	**aai**	**aau**
u	**ui**	
o		ou
o	**oi**	

Figure 1-5 Cantonese Diphthongs (Based on Bauer and Benedict 1997)

In “Modern Cantonese Phonology”, work done by Li Xingde in 1985 is presented (Bauer and Benedict 1997). Figure 1-6 is based on Li’s work and shows the average duration of 5 types of finals in milliseconds.

Type of Final	Average Duration
V:	280 ms.
V:C_{nasal}	196 ms.
V:C_{stop}	159 ms.
VC_{nasal}	99 ms.
VC_{stop}	83 ms.

Figure 1-6 Average Duration of 5 Types of Finals (from Bauer and Benedict 1997)

Average duration for any given long vowel is 212 milliseconds ((280 + 196 + 159) / 3). The average duration for any given short vowel, however, is only 91 milliseconds ((99 + 83) / 2). Interestingly, the average duration for a long vowel with no coda is 280 milliseconds, whereas the duration of an unreleased stop is quite shorter at 159 milliseconds. Note that data for short vowels with no coda are not represented in Figure 1-6.

1.3 Cantonese Tones

1.3.1 Definition of Tone

A ‘tone’ is simply the movement of pitch (or fundamental frequency) over time. The greater the number of vibrations in a speaker’s vocal cords, the higher the resultant fundamental frequency. Pitch is measured in Hertz (Hz) and signifies the number of vibrations per second. A speech sample with a fundamental frequency of 200 Hz, for example, means that the vocal cords opened and closed 200 times in one second.

Although tones are lexical in Cantonese, Cantonese words do have a consistent range of acceptable pitches and contours. Acceptable pitches and contours are different for each speaker and for each syllable's context. It has been shown, for example, that a woman's low tone may be pronounced at the same pitch as a man's high tone, or indeed higher (Matthews and Yip 1994). The absolute values of fundamental frequencies are not as important as the relative heights compared to nearby syllables.

Not only is the pitch variation great among different speakers, but it also exists within an individual's speech. Duplicating one's tonal contour is nearly impossible. Fundamental frequency is easily affected by one's physiological condition, speaking style, and emotional status (Lau et al. 2000b). If a speaker repeatedly speaks the same word, the temporal pitch movement will be similar, yet different, in each instance due to uncontrollable speaker variability.

1.3.2 History of Chinese Tones

Chinese historical phonology is based on the Qièyùn dictionary (compiled in AD 601), which is practically identical to Middle Chinese (Norman 1988). Middle Chinese contained 8 tonal categories as shown in Figure 1-7 divided into two subcategories: *yīn* and *yáng*. The distinction between the two categories in Middle Chinese was that *yīn* contained syllables with voiceless initials, whereas *yáng* contained syllables with voiced initials. Each category had four tones: a level tone, a rising tone, a falling tone, and an entering (or checked) tone.

	ping(level)	shăng(rising)	qù(falling)	rù(entering)
Yīn(voiceless)	1	3	5	7
Yáng(voiced)	2	4	6	8

Figure 1-7 Tonal Categories of Middle Chinese (Based on Norman 1988)

Most modern Chinese dialects have merged categories over time. Mandarin, for instance, has merged categories #3 and #4, #5 and #6, and has lost the rù category entirely (#7 and #8). Cantonese is one of only a few dialects that preserve all categories from Middle Chinese (Figure 1-8). In fact, Cantonese holds the rare distinction of being the only dialect in which a tonal category has split (i.e. #7 into #7a and #7b) (Norman 1988).

	ping(level)	shăng(rising)	qù(falling)	rù(entering)	
Yīn(voiceless)	1	3	5	7a	7b
Yáng(voiced)	2	4	6	8	

Figure 1-8 Tonal Categories of Modern Cantonese (Based on Norman 1988)

It should be noted, however, that the tonal categories have been preserved but not the actual tonal contours. Words in category #2, for example, were presumably pronounced with a level tone in Middle Chinese, but now possess a falling contour in modern Cantonese.

Tonal Category	Tonal Contour
1	53
2	21
3	35
4	24
5	44
6	33
7a	55
7b	44
8	33

Figure 1-9 Contours of Cantonese Tones as Spoken in Gwongzhou (from Norman 1988)

Figure 1-9 shows the values used to represent contours based on Chao's (1947) 5-point scale in which a '5' represents a high tone, a '1' represents a low tone, and the other values fall somewhere in between. The first digit represents the beginning position for the tonal contour; the second represents the ending position.

1.3.3 Number of Tones

It is very unclear and debatable how many tones there are in Cantonese. Linguists have argued for 6, 7, 9, 10, or even 12 tones in Cantonese. Pedagogical texts typically use 6 or 7 tones. Ching et al. (1994) has stated that there are "nine tones in Cantonese that are distinguishable from their pitch variation, loudness, and length". The correct number of tones there are in Cantonese, of course, depends on several variables including how one defines a tone.

One consideration is whether or not one views the high-level tone and high falling tone as separate or merged. These two tones are preserved in the province of Canton, but are merged by most speakers in Hong Kong (Bauer and Benedict 1997). Even in Hong Kong, a very strong high falling tone is maintained on the sentence final particles *sin* 'first', *tim* 'also', and on contracted numbers such as *sa'a yi* 'thirty-one'.

It is not surprising that many adopt this policy of merging, as the two tones are very similar, as well as the fact that most linguistic research is based on Cantonese as spoken in Hong Kong. Merging of the high level and high falling tones is also common in pedagogical contexts to simplify the already complex tonal system for learners of Cantonese.

Another consideration is whether or not the three entering (or checked) tones are viewed as distinct tones. In terms of categories, they are distinct from the non-entering categories primarily in duration. In terms of tonal contour, however, there is no difference between the three entering tones and their non-entering equivalents.

Because the pitch heights of the entering tones correlate with non-entering tones, it is no surprise that many do not consider them as tones at all. If one considers duration as distinctive for a tonal category, however, or if one would like to follow traditional Chinese classifications, then the three entering tones are included in the Cantonese tone system.

Bauer and Benedict (1997) propose two additional tones. These two tones result from a process called *bin3jam1* 'changed sound'. The first is a high level tone with a Chao tone value of '55' which is produced due to a process in which a high falling tone changes to a high level tone under certain contexts. The second *bin3jam1* tone has a Chao tone value of '25' and is produced when a tone undergoes a phonetic or morphological transformation (Bauer and Benedict 1997).

It is well documented, however, that there are 9 citation tones in Cantonese. In other words, when a native speaker of Cantonese is prompted with a Chinese character in isolation, only 9 tones are possible. The high falling example listed above only occurs in a specific context and therefore does not qualify as a citation tone. For the same reason, the *bin3jam1* tones do not qualify.

Unlike Mandarin, Cantonese does not have a well-known numbering system to use as a standard to represent the various tones. This thesis adopts the LSHK numbering system as developed by the Chinese University of Hong Kong (Lo et al. 1998).

LSHK romanization places a number 1 through 9 following a syllable to represent the tone associated with the syllable. Figure 1-10 provides a description for each of the 9 tones as well as the Chao tone values presented earlier. The reader should be careful not to confuse the tone numbers 1 through 9 with the tonal categories in Middle Chinese, as there is no correlation.

Tone	Description	Chao Value
1	High Level	55
2	Middle Rising	35
3	Middle Level	44
4	Low Falling	21
5	Low Rising	24
6	Low Level	33
7	High Level (entering)	55
8	Mid Level (entering)	44
9	Low Level (entering)	33

Figure 1-10 The 9 Tones of the LSHK Romanization System

1.3.4 Modification of Tonal Contours

There are several ways in which a canonical tone may be altered. Tone sandhi, tone change, tone coarticulation, and tone declination are four terms that describe these changes, yet are easy to confuse when dealing with Cantonese.

Tone sandhi refers to a systematic change of tone because of a specific tonal context. Some dialects of Chinese possess very complex system of tone sandhi. In Cantonese, the only case of tone sandhi is the change of a high falling tone to a high level tone when preceding another high tone. If one considers the high level and high falling tones as merged, however, then tone sandhi does not exist in Cantonese since the sole tone sandhi rule above has no application.

Cantonese has a very rich system of tone change. Tone change refers to a process in which a tone changes for morphological or semantic reasons. Such change is lexicalized and only occurs in specific words or lexical contexts. For example, the second adjective in adjective duplication becomes tone #2 such as *maan6 maan6 haang4* ‘take it easy’ becoming *maan6 maan2 haang4*. In kinship terms, which often consist of duplicated characters, it is common for the first of the duplicated syllables to become tone #4 as in *go1 go1* ‘older brother’ becoming *go4 go1*.

Tone coarticulation is a phonological process where a tone is altered due to its surrounding tonal context. A low tone following a high tone, for example, is likely to start with higher pitch than normal due to the speaker’s tendency to smooth the transitions between tones. Tone coarticulation has a bearing on results in this thesis, since the data used has been affected by tone coarticulation.

A final type of tone change is tone declination. Tone declination occurs over the course of an entire speech utterance. When a Cantonese phrase is spoken, speakers tend to slightly lower the heights of the tones. High tones, for example, are spoken at lower and lower pitch heights throughout a speech utterance, thus exhibiting sentential tone declination or down-drift (Lee et al. 2002a).

A sound understanding of Cantonese tones and Cantonese phonology is an essential precursor for doing tone recognition work. The concepts presented above are central to this thesis and will be referred to frequently in future sections.

1.4 Speech Recognition

The term ‘speech recognition’ refers to the design and implementation of computer algorithms to recognize the linguistic content of a spoken utterance. For a detailed overview of speech recognition, refer to the work of Rabiner et al. (1996). This section briefly discusses several speech recognition issues important to this thesis.

1.4.1 Introduction to Speech Recognition

Speech technology has great potential in Chinese communities due to the complex nature of their writing system. For instance, there is a dire need for an efficient method to input Chinese into a computer. It has been noted that learning time for users of the Chinese keyboard systems ranges from 15 days to 104 days (Wagner et al. 1986). Speech recognition is a technology that can solve many problems, such as the Chinese input issue described above.

An integral part of the Chinese speech recognizer is the tone recognizer. Many current Chinese recognizers do not use tone information even though it has been shown that inclusion of tone information can greatly enhance recognition accuracy. Even though results from their tone recognizer contained mistakes, Lau et al. (2000b) reported that their recognition rate of 75.4% was improved to 76.6%, or a 4.9% relative reduction in error rate. Subsequent research has also shown that “with perfect tone information, an improvement of 11.28% and 11.09% is achievable for Mandarin and Cantonese respectively” (Lau et al. 2000a). A tone recognizer could benefit from knowledge gained from a parallel phoneme recognizer; the method utilized in this thesis, however, does not contain any such phoneme recognition capabilities.

1.4.2 Scope

Previous research in tone recognition has fallen under four broad categories: monosyllabic, disyllabic, polysyllabic, and continuous. Monosyllabic and continuous recognition have received the most attention. The scopes of monosyllabic, disyllabic, and polysyllabic recognition include 1, 2, and many syllables respectively. The scope of continuous tone recognition is much larger and much more complex as its scope covers entire speech utterances.

Because there has been very little tone recognition research performed for Cantonese tones, monosyllabic tone recognition was chosen as the scope for this thesis as it is the most simple and straightforward. Simplification of the recognition task to a single character is particularly appropriate since each Chinese syllable is lexical and is the building block of the language. Liu et al. (1989) has reported that 23% of words in Chinese are monosyllables (62% are disyllables, 6% are trisyllables, and 9% are more than three syllables). Because Cantonese tone recognition in current continuous speech recognizers have much room for improvement, it is hoped that this thesis will serve as a reference for future speech recognition approaches using more complex scopes, such as continuous speech.

Only research in monosyllabic tone recognition is presented here. Readers interested in disyllabic recognition should refer to the work of Zhang and Hirose (1998). For polysyllabic recognition, refer to the work of Wu and Inoue (1991). Much research has been done in continuous speech, both in Mandarin (Wang et al. 1994; Chen and Wang 1995; Ma 1987; Zhang and Hirose 2000; Lau et al. 2000a; Gao et al. 2000), as well as in Cantonese (Wong et al. 1999; Chow et al. 1998; Ng et al. 1996; Lau et al. 2000a, 2000b; Gao et al. 2000).

1.4.3 Speaker Dependency

Speech recognition systems typically come in two kinds: speaker-dependent and speaker-independent. Speaker-dependent systems require users to provide speech samples before using. Providing speech data before use (also called ‘training’) allows the recognizer to better classify speech utterances through its increased knowledge about the data it is trying to recognize. Speaker-independent systems, on the other hand, do not require the user to initially train the system, and is therefore aimed at the ideal speaker rather than being patterned toward an individual speaker.

Because the training process often requires many hours of prerecorded speech, speaker-independent systems are usually preferred in applications where training is not feasible, such as in telephone-based applications. Speaker-dependent systems, however, are generally more accurate, and outperform speaker-independent systems by a factor of two to three (Huang and Lee 1993). Experiments in this thesis were performed using both speaker-dependent and speaker-independent systems.

1.4.4 Difficulties in Comparisons

The following section contains a history of relevant research to the Chinese tone recognition task. The reader should keep in mind that it is very difficult to judge the merits of an approach solely by the net accuracy attained. Experiments differ greatly in complexity, scope, corpora used, subjects, and even language. Comparison between approaches on basis of net accuracy alone will likely give a false impression as to the approaches' effectiveness. Due to the scarcity of research on Cantonese tone recognition, tone recognition research in other languages such as Mandarin and Thai is also provided. A summary of research is presented at the end of the section in Table 1-1.

1.5 Tone Recognition

1.5.1 Neural Networks

In the 1980's neural networks received significant attention in speech recognition research. Neural networks have been shown to be especially well suited for small-vocabulary recognition tasks (Huang et al. 2001). The term 'multi-layer perceptron' (MLP) refers to a commonly-used type of neural network. The layers in an MLP refer to the layers of neurons in its design. Neural networks consist of an input layer, an output layer, and optional hidden layer(s) in between. An example of a 3-level feedforward neural network is shown in Figure 1-11.

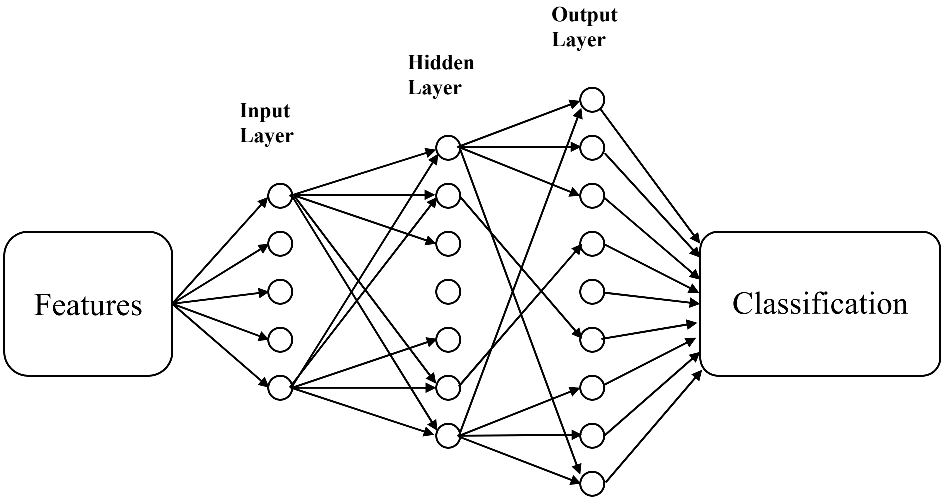


Figure 1-11 3-Layer Feedforward Neural Network Classifier

The number of nodes in the input layer reflects the number of input features. Each node in the output layer represents one of the possible classifications. The number of hidden layers, as well as the number of neurons in each, is flexible and is set specifically for each task. The more neurons the hidden layer possesses, the more accurate the system will be (up to the point of over-training), and the more training data is required. An increase in the number of hidden layers is typically reserved for more complex classification tasks and requires much more training data.

Although Figure 1-11 does not show it, every node in a neural network is connected to every node in the layer before and after it, and each path has an associated probability. When a new set of features is presented to the neural network, the classification is determined by finding the output node with the highest degree of activation.

Prior to this thesis, the only published work on monosyllabic Cantonese tone recognition is from Tan Lee and P.C. Ching from the Chinese University of Hong Kong (Lee and Ching 1997, 1999; Lee et al. 1993, 1995). The first step in their neural network approach involved detecting the voiced portion of a syllable and dividing it into 16 even segments. A pitch was then determined for each segment. The input for their neural network consisted of 5 normalized features: initial pitch, final pitch, rate of pitch increase, duration (of voiced portion), and energy drop rate. This set of features, or feature vector, was then entered into a neural network with one hidden layer and an output layer consisting of 9 nodes, one for each of the nine possible tones. The number of hidden neurons used was 25 for the single-speaker system, and 35 for the multi-speaker system. Recognition accuracy of 89.0% and 87.6% was achieved for single-speaker and multi-speaker recognition respectively.

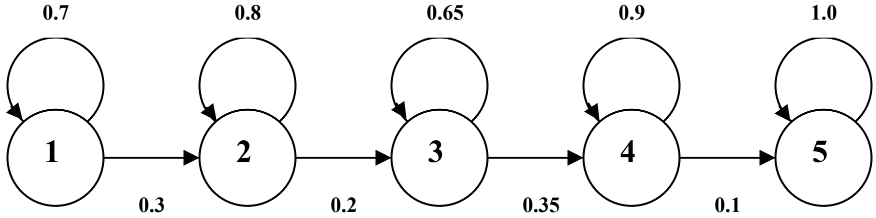
Neural networks have also been applied to Mandarin tone recognition by Wang et al. (1988, 1991). Wang's system used adaptive weights, a procedure where the probabilities between nodes can be altered during the recognition process. Recognition accuracy of 92.5% was obtained for Mandarin four-tone recognition. After successive training, accuracy was improved to 97.5%.

Another neural network approach for the Mandarin tone recognition task was attempted by Chang et al. (1990) using a back-propagation algorithm. Back propagation refers to a process where errors in classification are reported back into the system, used to modify probabilities in the hidden layer, and thus reducing error rate over time. The voiced portion of each speech item was divided into three non-overlapping segments. 10-dimensional feature vectors were then formed by extracting energies, means, and slopes from the tonal contours, as well as the duration of the voiced portion. The neural network design had 10 input neurons and 4 output neurons, one for each of the Mandarin tones. Various numbers of hidden layers (0-3) were attempted as well as various numbers of neurons per layer. Best results were achieved with one 12-neuron hidden layer, obtaining 93.8% accuracy.

1.5.2 Hidden Markov Models

The use of hidden Markov models (HMMs) is currently the best-performing speech recognition approach (Lippmann 1990; Cox 1990; Ostendorf 1996). For this reason, research in HMMs far outweighs the research in any other area of speech processing (Rabiner 1989). HMMs work well for word-spotting (Rose 1996) and dominate the continuous speech recognition field for their acoustic modeling ability (Ostendorf 1996).

An HMM is a statistical model that uses a finite number of states and associated state transitions to model both temporal and spectral variations of signals (Rabiner et al. 1996). There are many variants of HMMs. One of the simplest is provided in Figure 1-12 showing a 5-state (numbered 1 through 5) left-to-right hidden Markov model.



$$\text{Transition Probability Matrix} = \begin{bmatrix} 0.7 & 0.3 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.65 & 0.35 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.9 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

Figure 1-12 A 5-State Left-to-right Hidden Markov Model (Based on Cox 1990)

Each state has one or more possible paths leaving from it with an associated probability. The sum of the probabilities must total 1.0 for each node. The first node in Figure 1-12, for example, has a 0.7 probability of remaining in the current state and a 0.3 probability of moving to the next state. Probabilities of the HMM model are typically represented in a transition probability matrix as shown in Figure 1-12. The first row and column of a transition probability matrix denote state #1; the second row and column denote state #2, and so on.

The basis of Markovian theory, which is used in HMMs, is that the likelihood of being in a given state depends only on the immediately prior state, and not on earlier states (O'Shaughnessy 2000:85). All that is needed to predict the future is to know the value of the present variable, but not past variables. If one wanted to know how many employees work for Company A, for example, knowledge of how many employees were there yesterday will give a highly accurate idea of how many are currently there. Knowledge of how many employees there were a week or a year ago would not be considered important.

HMMs are called 'hidden' because an outside observer can only observe the final output, not knowing the actual state sequence within the HMM. In HMM-based tone recognition, each individual tone is typically modeled with its own HMM. Classification is then performed by computing the probability of generating the test utterance with each tone model and selecting the tone associated with the highest scoring HMM (Chen et al. 1987). Many different algorithms exist to perform HMM traversal in the calculation of scoring functions.

Several researchers have applied hidden Markov models to tone recognition of Mandarin syllables. Chen et al. (1987) have applied HMMs using Baum's forward-backward algorithm, in which HMM traversal occurs in both directions (i.e. from the beginning, moving left to right, and from the end, moving right to left). With an HMM created for each tone, 98% accuracy was reported for recognition of 35 syllables. 96% accuracy was reported in an informal experiment of recognizing Mandarin digits.

Wang and Iso-Sipilä (2002) applied left-to-right HMMs without state transition matrices to Mandarin tone recognition. 3 states were allocated to initial models, and six states for final models due to the relative lengths of the initial and final parts. Good results were obtained in this approach even when the complexity of the task was reduced by 60%. In their 'Elephant' system, speaker-independent accuracy was 87.9%. Using speaker-adaptation, 89.7% accuracy was obtained using their 'Tiger' system.

Lin et al. (1993) also used an HMM-based approach for Mandarin tone recognition. Interestingly, they did this without using pitch information as inputted features. Their speaker-dependent results were surprisingly good, at 94.85%, despite the small amount of data used (only 2 speakers).

Apichat Tungthangthum (1998) applied HMMs to the recognition of tones in Thai. HMMs were trained for each of the five tones in Thai, with classification output being assigned to the tone associated with the highest scoring HMM. The experiment was speaker-dependent with data coming solely from one speaker. 90% accuracy was reported.

1.5.3 Vector Quantization

Vector quantization was introduced by Shannon in the late 1950's (Shannon 1960). Vector quantization is a process that maps a sequence of continuous vectors into a digital sequence (Gray 1990). In other words, it is a data compression principle that finds the set of vectors that represent an information source with minimum expected distortion (Burton 1985). It is also an effective method for finding cluster centers in data (Jelinek

1997). Quantization of a single signal value is referred to as scalar quantization, whereas vector quantization refers to quantization of multiple signal values (Huang et al. 2001).

When large amounts of data are being used, it is favorable to skillfully select representative samples from the entire population. Each item selected as a representative is called a 'codeword', with the entire collection of codewords comprising a 'codebook'. Using vector quantization eliminates the need for involving each training item in calculations, thus reducing time needed for classification. For this reason, it is a common technique used in speech recognition tasks involving large amounts of data.

Guan and Chen (1993) applied vector quantization to Mandarin tone recognition. They represented their data with a 2-dimensional decision vector. Results from their speaker-independent experiments yielded 98.76% for isolated Mandarin syllables.

1.5.4 Fuzzy Sets

Fuzzy set theory was first postulated by Lotfi Zadeh (1965). In this paper, he mentions that 'imprecisely defined "classes" play an important role in human thinking, particularly in the domains of pattern recognition, communication of information and abstraction'. The idea behind fuzzy logic is that as systems become more and more complex, difficulty in making precise categorizations increases.

Linguistics is full of imprecise information such as 'pretty tall' or 'not so tall', which are not very specific, yet contain enough information to express knowledge of the state of the matter (Hernández-Ábrego 2000). The strength of fuzzy logic is that it can simultaneously handle numerical and linguistic knowledge, and is the only method able to do so in a unified and mathematical manner (Mendel 1995). Once a fuzzy logic system is developed, fuzzy functions are then produced for classification. One setback in using this approach is that it requires careful understanding of fuzzy logic as well as sound knowledge of the linguistic principles involved (Mendel 1995).

Fuzzy sets were applied by Xu and Lee (1992) to Mandarin tone recognition. The four Mandarin tones were considered as four separate fuzzy sets which describe the four pattern classes. The membership functions of the four fuzzy sets were constructed through analysis of the tonal contours of the tones. In a speaker-dependent experiment, accuracy of around 99.5% is reported.

1.5.5 Decision Trees

Using decision trees is a comparatively simple approach to classify speech data. A decision tree is essentially a flow-chart of decisions which ultimately leads to a classification. When a new speech sample is presented, the classification is decided upon by going through the decision tree, with branches being selected depending on the answer to various questions. One advantage of using a decision tree is that it is specifically geared toward the current task and is easily altered to improve results. On the other hand, it is not portable to other tasks.

Chang and Yang (1986) applied this approach to Mandarin tone recognition. Accuracy of 90.2% was obtained, despite using a very small training set (10 speakers speaking 100 digits each).

1.5.6 Hybrids

A hybrid approach is simply a combination of two or more of the above approaches. A common hybrid approach is to use vector quantization to simplify data and then to use the resultant codewords or codebooks as input into the classification portion of the recognizer. Each of the experiments sketched below utilizes this technique.

After using vector quantization on their data, Yang et al. (1988a, 1988b) use HMMs to model Mandarin tones. They reported 98.33% accuracy in the speaker-dependent case, and 96.53% in the speaker-independent case.

After vector quantization, Liu et al. (1989) matched states of their HMMs with the three or four turning points of a Mandarin syllable's tonal contour. They found that using multiple models for each tone gave better recognition rates. Codebook size was reported to work best when set to 32. Recognition accuracy of 97.9% was observed for isolated monosyllabic words (92.9% for disyllabic and 91.0% for trisyllabic).

Zhou and Imai (1996) utilized a different hybrid approach. After using vector quantization, they used a combination of vector quantization and multi-layer perceptron (a type of neural network) to classify Mandarin tones. Results were very high at 99.82%.

Table 1-1 summarizes the research presented in this chapter. Note the vast majority has used neural networks, HMMs, vector quantization, or a hybrid of these three methods. It will be shown that the new approach used in this thesis is extremely successful in recognizing tones. The next chapter details the methodology used in this thesis.

Method	Author(s) and Year	Speaker-Dependent	Speaker-Independent
Neural Network (Cantonese)	**Lee et al. 1993,1995**	**89.0%**	**----**
Neural Network	Wang et al 1988, 1991	97.5%	92.5%
Neural Network (w/ back propagation)	Chang et al. 1990	---	93.8%
HMM + differential coding scheme	Chen et al. 1987	96.0%	---
HMM (w/o pitch features)	Lin et al. 1993	94.9%	---
HMM (w/ 60% lower complexity)	Wang and Iso-Sipilä 2002	89.7%	87.9%
HMM (Thai)	Tungthangthum 1998	90.0%	---
Vector Quantization (VQ)	Guan and Chen 1993	---	98.8%
Fuzzy Sets	Xu and Lee 1992	99.5%	---
Decision Tree	Chang and Yang 1986	---	90.2%
VQ/HMM	Yang et al. 1988a, 1988b	98.3%	96.5%
VQ/HMM	Liu et al. 1989	---	97.9%
VQ/MLP	Zhou and Imai 1996	99.8%	---

Table 1-1 Summary of Monosyllabic Tone Recognition Research (Cantonese in bold)

2 Methodology

This chapter contains a step-by-step description of the methodology used in this thesis. The process begins with files containing speech utterances. Each file is processed with the last step being the assignment of tone.

Figure 2-1 provides a general overview of the tone recognition process.

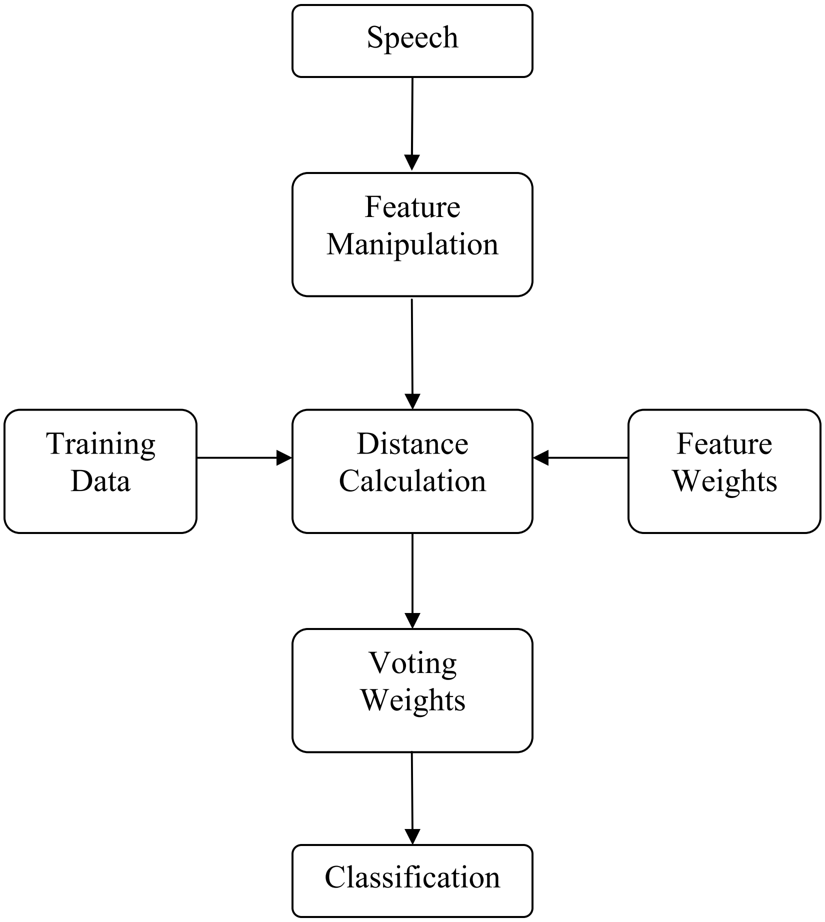


Figure 2-1 Overview of Tone Recognition Process

2.1 Corpus Selection

In this work, I used the Chinese University Syllable Corpus (CUSYL version 1.0). It was the natural choice as it is the only publicly available spoken-language corpus of Cantonese syllables. It is, however, well suited for the Cantonese tone recognition task. CUSYL, a subset of the larger CUCorpora, is a corpus of single syllables produced and distributed by the Speech Processing Group of the Digital Signal Processing Laboratory at the Chinese University of Hong Kong. The corpus was specifically designed to cover the entire set of valid Cantonese syllables, including alternative and colloquial pronunciations (Lee et al. 2002).

The database contains speech from 2 male (cs02m, cs04m) and 2 female (cs01f, cs03f) speakers, each speaking the entire set of about 1,800 Cantonese syllables. Table 2-1 displays the syllabic inventory of the corpus. Table 2-2 and Table 2-3 provide the tonal distribution. All data is manually verified using the LSHK phonemic transcription format also developed at the Chinese University of Hong Kong.

Total Number of Syllables	1801
Number of Distinct Tonal Syllables	1388
Number of Distinct Base Syllables	637

Table 2-1 Syllabic Inventory of CUSYL (Lee et al. 2002)

Tone	1	2	3	4	5	6
Count	423	319	351	247	136	325

Table 2-2 6-Tone Distribution in CUSYL (Simplified form of Table 2-3)

Tone	1	2	3	4	5	6	7	8	9
Count	328	319	253	247	136	217	95	98	108

Table 2-3 9-Tone Distribution in CUSYL (Lee et al. 2002)

After being prompted in Chinese characters, the speakers spoke into a head-mounted microphone. Due to undesirable syllable lengthening in monosyllabic speech, the prompts also included "carrier words", which were then deleted from the data (Lo et al. 1998). Although the files themselves contain only one syllable, they should not be considered isolated syllables per se, since coarticulation effects are present on both ends of the tonal contour.

Subjects were recorded using a Shure SM10A dynamic cardioid head-mounted microphone. The data was passed through a mixer to the DAT recorder for real-time A/D conversion at 48kHz. The data was then reduced to 16kHz with a DATLink DSP firmware and sent through a SCSI interface to the computer's hard disk (Lo et al. 1998). A summary of recording environs is shown in Table 2-4.

Microphone	Shure SM10A Dynamic cardiod type
Mixer	Mackie 1202 VLZ
A/D Conversion	Sony PCM 2700A at 48kHz
Downsampling	DAT Link default downsampling setup

Table 2-4 Summary of Recording Environment

2.2 Feature Manipulation

2.2.1 Feature Extraction

The first step in processing a file is to extract the desired features from the speech sample. There were two main feature extraction methods used in this thesis to extract fundamental frequencies (F_0). Both algorithms were implemented using tools included in the "Speech Filing System" (SFS), a freely available suite of speech processing tools available from the University College London (ftp://ftp.phon.ucl.ac.uk/pub/sfs). The first is an autocorrelation algorithm; the second is a cepstrum algorithm.

Relevant documentation (Huckvale 2000) describes the autocorrelation method (*fxac*) as a three-step process: (i) cubing waveform sample values, (ii) autocorrelation, and (iii) voicing & fundamental frequency decision. Output values are estimated in 25ms windows with a repetition time of 5ms (Huckvale 2000). Examination of the code suggests that the search range is restricted to 80-400 Hz (de Cheveigné 2001).

According to Huckvale's documentation (2000), the cepstrum algorithm (*fxcep*) first decomposes the waveform using a 512-point Fast Fourier Transform (FFT) on 40 millisecond windows of input speech, and then calculates the log spectrum. An FFT of this result then provides the cepstrum. Noll rules (Noll 1967) are then implemented to find out if the input is voiced, and if so, a fundamental frequency value is determined (Huckvale 2000). Examination of the code suggests that the search range for this calculation is limited to 67-500 Hz (de Cheveigné 2001).

2.2.2 Zero-Removal

Results from both the autocorrelation and cepstrum methods consist of a series of data elements. Each element is either a positive, whole number indicating the F_0, or a zero-value denoting an absence of F_0. Because F_0-values are only obtained from voiced portions of speech, and only voiced speech carry tone information, it was decided that zero-features (i.e. features with a value of '0') would be removed. Pre-syllable, postsyllable, and even mid-syllable zero-values, if any, are removed in this process. It is assumed that a mid-syllable zero-value does not contain pertinent tone information and is thus disregarded as an error in the feature extraction process. A slight gap due to the loss of a feature is more desirable than inclusion of an erroneous zero-value, which would adversely affect future calculations. An example of a feature vector both before and after the Zero-Removal step is provided in Table 2-5. Notice that the 3 leading zeroes have been truncated.

Before Zero-Removal	*0 0 0 104 104 205 205 205 200 195 192 188 183 181 179 177 175 175 173 170 168 163 163 163*
After Zero-Removal	*104 104 205 205 205 200 195 192 188 183 181 179 177 175 175 173 170 168 163 163 163*

Table 2-5 Data before and after Zero-Removal step for a Cantonese syllable

2.2.3 Feature Selection Algorithm

In order for the methods proposed in this paper to work, each input must contain an equal number of features. It is still uncertain what value is the ideal number of features to use for tone recognition. Prior methods have tried such values as 10, 16, or even 80 (Zhang et al. 2000, Lee et al. 1993, Chen et al. 1987). A set of 16 features was selected as the best option for this thesis to facilitate comparison with Lee et al. (1993) as well as provide a benchmark value. This thesis also performs experiments using 8 features in an attempt to find a less computationally expensive value.

In cases where the input does not contain the required number of features (only occurs in 16-feature experiments), zeroes are added to the beginning of the feature vector to meet the vector length requirement (Table 2-6 and Table 2-7). In both tables, *N* refers to the required number of features (e.g. *N* = 16 in Table 2-6), whereas *e* refers to the total number of features in the input (e.g. e = 5 in Table 2-6). In most cases, however, the input possesses more than the minimal requirement of features. In such cases, the first and last features are selected. Remaining features are selected such that they are maximally equidistant from each other. The equations used for this process are shown in Equation 2-1.

Before Feature Selection	*246 246 246 246 242*
After Feature Selection	*0 0 0 0 0 0 0 0 0 0 0 246 246 246 246 242*

Table 2-6 Data before and after Feature Selection step when *e* < *N*

Before Feature Selection	*104 104 205 205 205 200 195 192 188 183 181 179 177 175 175 173 170 168 163 163 163*
After Feature Selection	*104 104 205 205 200 192 188 183 179 177 175 173 170 163 163 163*

Table 2-7 Data before and after Feature Selection step when *e* ≥ *N*

$$f_i = \begin{cases} 0 & for \quad i = 1 \ to \ (e-1) \\ f_i & for \quad i = e \ to \ N \end{cases} \qquad\qquad f_i = \text{int}_{i=1 \, to \, N}\left(i + \frac{e-1}{N-1} \right)$$

Equation 2-1 Feature Selection Algorithm for *e* < *N (left) and e* ≥ *N (right)*

2.2.4 Feature Extraction Variations

Although the insertion of zeroes in the procedure mentioned above helps meet requirements for the number of features, it does not retain tonal contours very well. As will be shown later, these zero-values greatly increase absolute values of scalar distances between two features being compared. In an attempt to minimize this effect, two variations to the above feature extraction algorithm were explored in this work. The first is called “Feature Stretching”; the second is “100-Addition”.

In the “Feature Stretching” method, features are “stretched” out to meet the number of features requirement. The resulting contour can be reasonably retained, especially in level tones as Table 2-8 illustrates. Note each of the 5 features are “stretched” out to meet the vector length requirement of 16 features. Tones with extreme contours, on the other hand, are lengthened, thus making strong pitch changes appear much gentler.

Before Feature Selection	*246 246 246 246 242*
After Feature Selection	*246 246 246 246 246 246 246 246 246 246 246 246 246 246 242 242*

Table 2-8 Data before and after Feature Stretching Algorithm

The algorithm is shown in Equation 2-2, is identical to the feature selection method outlined above in the case where $e \geq N$. The sole difference lies in that the Feature Stretching algorithm is used for the calculation of every feature.

$$f_i = \operatorname{int}_{i=1\,to\,N}\left(i + \frac{e-1}{N-1}\right)$$

Equation 2-2 Feature Stretching Algorithm

It was observed that the zeroes added during the “Zero-Addition” process introduced a severe penalty for feature vectors without enough features. In the “100-

Addition" method, values of '100' are added, instead of zero-values, in each case where there is an insufficient number of features (Table 2-9). Using "Feature Stretching", the new feature takes on the value of the nearest feature; using "Zero-Addition", a zero is added.

Before Feature Selection	*246 246 246 246 242*
After Feature Selection	*100 100 100 100 100 100 100 100 100 100 100 246 246 246 246 242*

Table 2-9 Data before and after 100-Addition Algorithm

The value of '100' was arbitrarily chosen because it is a value roughly in the middle of the two. The hypothesis is that inserting a value of 100 would help smooth the contours and lessen the dramatic effect of zero-addition. The algorithm employed is shown in Equation 2-3, and is identical to the zero-addition algorithm except for the addition of 100 instead of 0.

$$f_i = \begin{cases} 100 & for \quad i = 1 \ to \ (e-1) \\ f_i & for \quad i = e \ to \ N \end{cases}$$

Equation 2-3 100-Addition Algorithm

2.2.5 Normalization

For speaker-independent experiments, normalizing feature vectors can have a very strong effect on recognition accuracy. Because each speaker speaks in a different frequency range with different distribution, normalization of feature vectors enhances the ability to compare data across speakers. Table 2-10 displays the mean (average value of all frequency values) and standard deviation that was calculated for each of the four speakers in the CUSYL corpus. As expected, the mean frequencies from the two female speakers (cs01f, cs03f) are noticeably higher than the male speakers'. Note also that the standard deviation for speaker cs03f is significantly higher than the others. This will be discussed in future sections.

	Mean (μ)	Standard Deviation (σ)
cs01f	201.3	38.8
cs02m	155.6	32.0
cs03f	220.9	49.3
cs04m	150.1	28.3

Table 2-10 Mean and Standard Deviations of Speakers in the CUSYL corpus

This thesis adopts the standard method of normalization by altering each feature by subtracting the mean (μ) and then dividing by the standard deviation (σ) as shown in Equation 2-4.

$$f_i = \frac{f_i - \mu}{\sigma}$$

Equation 2-4 Normalization Formula

The standard method of computing the mean is also used. Equation 2-5 shows the formula used to calculate the mean as well as the standard deviation formula used during normalization.

$$\mu = \frac{\sum_{i=i}^{N} f_i}{N} \qquad \sigma = \sqrt{\frac{\sum_{i=1}^{N} (f_i - \mu)^2}{N}}$$

Equation 2-5 Formulas to Calculate the Mean (left) and Standard Deviation (right)

This formula employs a less commonly used method of calculating standard deviation such that the denominator is *N*, rather than the usual *N-1*. Choosing a denominator of N is typical when measuring distribution whose mean is known a priori (Press 1992:611).

An example of this transformation can be seen in Table 2-11. Features no longer represent actual frequency values, but rather a relative value. A value of 0 would represent a frequency value equal to the speaker's mean frequency. A positive value indicates how many standard deviations the feature is above the mean; a negative value (none shown in Table 2-11) signifies how many below.

Before Normalization	*271 266 266 262 258 262 262 258*
After Normalization	*1.7955 1.6667 1.6667 1.5636 1.4604 1.5636 1.5636 1.4604*

Table 2-11 Data before and after feature normalization

The normalization process described above is essential in speaker-independent experiments, as normalization allows data from different speakers to be directly comparable. As will be shown later, normalizing of data has a profound effect on recognition accuracy.

2.2.6 Feature Trimming

Fundamental frequencies can only be extracted on voiced portions of speech. Because syllable initials are unvoiced in Cantonese, they carry no tone information. Subsequently, the tone recognition task for Cantonese is limited to data from the rime portion of the syllable: the vowel and the optional nasal coda. Data from the peripheries, especially from nasal codas, tend to be less predictable. It is unclear, however, which features from voiced regions are detrimental to recognition. Hengjie Ma (1987) mentions that he “cut small parts of the pitch contour near its endpoints in order to avoid the unfavorable influence”. Liu and Wang (1988) arbitrarily select the first and last 1/8 portion to discard. It is also common to consider tonal information from the vowel only, discarding information from nasal codas. Chen (2001) goes further to suggest that data from only the main vowel (not including glides) should be used.

The feature extraction methods described above are blind to phoneme information. In other words, fundamental frequencies are collected from any voiced speech data, whether it is vowel, nasal, or other. Limiting the task to consider only vowel data is impossible without integrating a phoneme recognizer. In an effort to increase accuracy, however, experiments were run in which features at the beginning and/or end of feature sets were ignored.

For this thesis, attempts were made to ignore the first and last feature (i.e. #1 and #8) as well as the first and last pair of features (i.e. #1-2 and #7-8) in 8-feature speaker-dependent experiments. In the 16-feature experiments, ignoring even more features was attempted. In addition to trying to ignore the outside 1 or 2 features mentioned above, experiments were performed in which outside 3 and 4 features were not considered. Attempts to ignore only the first feature, or two, or three were also tried (Table 2-12).

	# of Features	Ignored Features					
Speaker-Dependent	8 Features	1-8	1-2,7-8	---	---	---	---
	16 Features	1-16	1-2,15-16	1-3,14-16	1	1-2	1-3
Speaker-Independent	8 Features	---	---	---	---	---	---
	16 Features	---	---	---	1	1-2	1-3

Table 2-12 Selection of Features to Ignore

Feature trimming is an important part of the procedure presented in this thesis, as it essentially decides which features in a feature vector carry the most crucial information for the recognition task. It will later be shown that trimming features has a noticeable effect on recognition accuracy. Once the features have been extracted and manipulated, they are entered into the tone recognizer. The underlying theory behind the tone recognizer used in this thesis is 'memory-based learning'.

2.3 Memory-Based Learning (MBL)

Memory-based learning is a machine-learning approach referred to as example-based, exemplar-based, instance-based, similarity-based, nearest-neighbor, or lazy learning. Memory-based learning is composed of a learning component and a similarity-based performance component (Daelemans et al. 1998a).

The learning component takes feature vectors extracted from files and stores them into memory. No abstraction or structuring is performed, nor are any explicit rules or generalizations applied. Each case stored in memory consists of a feature vector with an associated classification label.

During the performance component, a previously unseen feature vector is presented and similarity is calculated between the new test case and each example stored in memory. New instances are then assigned the classification of the closest item(s) stored in memory. In cases of ambiguity, when several classifications are assigned to identical input, the most probable solution, or a probabilistic random choice, is made (Daelemans 1998).

Van den Bosch (1999a) notes that memory-based learning has attained adequate to excellent generalization accuracies on complex tasks as different as hyphenation, semantic parsing, part-of-speech tagging, morphological segmentation, and word pronunciation (Daelemans and van den Bosch, 1992; Cardie, 1994,1996; Daelemans et al., 1996; van den Bosch, 1997). A similar type of analogical reasoning has been applied in other areas as well (see Skousen 1989, 1992; Skousen et al. 2002).

Veenstra (1998) observes that the assumption underlying memory-based learning is that the use of stored experience outperforms the application of knowledge (such as rules or decision trees) abstracted from experience. Stochastic, rule-based approaches abstract rules from earlier experiences to apply to future test cases. It has been shown, however, that pure memory-based learning algorithms obtain better results in experiments such as grapheme-phoneme conversion and word pronunciation (van den Bosch 1999a, Veenstra et al. 1998).

The primary difference between lazy learning and the rule-based, or ‘eager’ learning, is that eager learning methods prune information during their abstraction processes. Eager approaches ignore exceptional, low-frequency items with the supposition that elimination of these items will aid overall accuracy. The strength of the lazy learning approach is that irregular, yet useful, information is not discarded. It is essentially task-independent, language-independent, and independent of expert knowledge, assuming the task can be described as a classification problem (Daelemans et al. 1998a). Furthermore, because rules are not abstracted, there is no need for rule ordering as needed by rule-based approaches (Daelemans 1998). Daelemans et al. (1999) explain in detail why forgetting exceptions can be harmful.

Memory-based learning, as detailed above, is one of the foundations of the new approach presented in this work. This thesis utilizes memory-based learning to categorize and classify Cantonese tone data.

2.4 TiMBL

The memory-based learning techniques presented in this thesis were implemented using TiMBL. TiMBL is a software package developed by the ILK group at Tilburg University and can be freely downloaded for research purposes from http://ilk.kub.nl/timbl_download.html. Although TiMBL was designed with linguistic tasks in mind, it can be applied to any classification tasks with symbolic or numeric features for which training data is available (Daelemans et al. 2001).

TiMBL, as a tool to implement memory-based learning, has been applied to many NLP tasks. Selected examples from TiMBL documentation (Daelemans et al. 2001) in the morpho-phonological areas are: classifying phonemes in speech (Kocsor et al. 2000); assignment of word stress (Daelemans et al. 1994); grapheme-to-phoneme conversion (van den Bosch and Daelemans 1993); predicting linking morphemes in Dutch compounds (Krott et al. 2001); diminutive formation (Daelemans et al. 2001); and morphological analysis (van den Bosch and Daelemans 1999).

Syntactico-semantic tasks at the sentence level include part-of-speech tagging (Zavrel and Daelemans 1999, van Halteren et al. 2001); PP-attachment (Zavrel et al. 1997); word sense disambiguation (Veenstra et al. 2000); subcategorization (Buchholz 1998b); phrase chunking (Tjong Kim Sang and Veenstra 1999); article generation (Minnen et al. 2000); shallow parsing (Buchholz et al. 1999); clause identification (Orasan 2000, Tjong Kim Sang 2001); and sentence-boundary detection (Stevenson and Gaizauskas 2000).

On the textual level, TiMBL has been used for information extraction (Zavrel et al. 2000) and spam filtering (Androutsopoulos et al. 2000). In the field of discourse and dialogue modeling, TiMBL has been used for shallow semantic analysis of speech-recognized utterances (Gustafson et al. 1999) and in error detection in spoken dialogue systems (Krahmer et al. 2001, van den Bosch et al. 2001).

TiMBL documentation goes further to say that while most work has been oriented towards language engineering applications, linguistic and psycholinguistic relevance of memory-based learning has been another focus. Work includes stress assignment in Dutch (Daelemans et al. 1994, Gillis et al. 2000); reading aloud (van den Bosch and Daelemans 2000); phonological bootstrapping (Durieux and Gillis 2000); and comparison to other analogical methods for linguistics (Daelemans et al. 1997).

In speaker-dependent experiments it is typical to divide a speaker’s speech data into two groups: training data and test data. In this work, 200 of each speaker’s 1800 total speech files were reserved for testing, with the other 1600 serving as training samples. Test cases were chosen by selecting every 9th file. Word accuracy was then computed by dividing the number of correct classifications by the total number of test cases (i.e. 200).

In speaker-independent experiments, data from 3 speakers were used for training with the remaining speaker (all 1800 files) being used to test. All permutations were performed using each speaker alone as the test case. Results were then computed by taking the average of the results from each of the four speakers. Because the test set for speaker-independent experiments was much larger (i.e. 1800 as compared to just 200) computational cost was increased; and due to hardware restrictions, less experiments could be performed in some cases.

2.5 Similarity Metrics

The similarity metrics used in this thesis define how to compute the likeness between two sequences of numbers (i.e. features). The three metrics offered by TiMBL are: Overlap, Modified Value Difference, and Numeric.

The Overlap metric for similarity is based on the number of features in two cases that match exactly. In other words, a feature with a value of '230' is as different from a value of '0' as it is different from a value of '229'. The formula for the Overlap metric can be found in Equation 2-6, where $\Delta(X,Y)$ is the distance between X and Y, represented by n features, w_i is a weight for feature i, and δ is the distance per feature (Daelemans 1999). Note that each of the metric formulas in this section, as well as the formulas presented in Chapter 2.6 and 2.8 are taken directly from TiMBL documentation (Daelemans et al. 2001).

$$\Delta(X,Y) = \sum_{i=1}^{n} w_i \delta(x_i, y_i)$$

where: $\delta(x_i, y_i) = 0 \; if \; x_i = y_i, \; else \; 1$

Equation 2-6 Overlap Metric Formula

The Overlap metric is restricted in application; it limits itself solely to exact matches between feature vectors. In contrast, the Modified Value Difference Metric (MVDM) was defined by Stanfill and Waltz (1986), and later added upon by Cost and Salzberg (1993). This new method does not see each value as equally dissimilar but rather classifies groups of features as more or less similar.

A certain application, for example, could consider the phonemes /m/ and /n/ to be more similar than /m/ is to /a/. The phonemes /m/ and /n/ are considered more similar because the phonemes have many articulatory features in common, such as ‘consonant’ or ‘nasal’. The phonemes /m/ and /a/, on the other hand, share very few features. Equation 2-7 shows the formula used to compute distances between two values V_1, V_2 of a feature using the Modified Value Difference Metric, where C_i is the class, and P is the conditional distribution (Daelemans et al. 2001).

$$\delta(V_1, V_2) = e\sum_{i=1}^{n} \left| P(C_i \mid V_1) - P(C_i \mid V_2) \right|$$

Equation 2-7 Modified Value Difference Metric Formula

TiMBL offers another metric specifically designed for numeric features. This third metric does not take into account the number of exact matches as the Overlap metric does, nor does it classify certain groups of features as more similar than others as the MVDM approach does. Distance is simply derived as the sum of feature distances, each in turn being the absolute value of the difference between feature values. The distance between vector A and vector B in Table 2-13 is 40 (5+5+5+5+5+5+5+5). The distance between vector A and C is 280 (70+60+50+40+30+20+10+0). In this case, vector A is much closer to vector B than it is to vector C. The Numeric metric is shown in Equation 2-8 (Daelemans et al. 2001).

Feature Vector	Features
A	200 200 200 200 200 200 200 200
B	195 195 195 195 195 195 195 195
C	130 140 150 160 170 180 190 200

Table 2-13 Feature Vector Values for Sample Calculation

$$\Delta(X,Y) = \sum_{i=1}^{n} w_i \delta(x_i, y_i)$$

where: $$\delta(x_i, y_i) = abs(\frac{x_i - y_i}{\max_i - \min_i})$$

Equation 2-8 Numeric Metric Formula

Because the features used in this thesis are numerical fundamental frequencies, the numeric metric was chosen. Test experiments in this thesis show that the numeric metric was most appropriate for this task. Results using the Overlap metric were considerably lower and MVDM was comparable to, but never outperformed the numeric metric.

Three similarity metrics have just been discussed: the Overlap metric, MVDM, and the Numeric metric. The Numeric metric is the best similarity metric for tone recognition and serves as a fundamental component of the new approach described in this thesis.

2.6 Feature Weighting

Without feature weighting, all features in the feature vector are assumed to be equally important in the classification task. However, particular features carry more pertinent information than others. For this reason, more informative features can be weighted more heavily than others to obtain better net accuracy. TiMBL provides five different feature-weighting methods to choose from: none, Information Gain, Gain Ratio, Chi-squared, and Shared Variance. In order to determine which feature-weighting method produced best results, experiments are performed using each of the five methods. Results using each of these methods will be shown in Chapter 3.4. For an overview of other feature weighting methods not described in this paper, refer to Wettschereck et al. (1996) and Aha (1998).

TiMBL documentation notes that for each of the feature weighting methods described below, numeric features are discretized into a number (default = 20) of equally-spaced intervals between the minimum and maximum values of the feature (Daelemans et al. 2001).

The first, and simplest, way to weight features is to weight all features equally. The net result is a simplification of Equation 2-9 (left equation) to a mere sum of feature distances, as shown in Equation 2-9 (right equation).

$$\Delta(X,Y) = \sum_{i=1}^{n} w_i \delta(x_i, y_i) \qquad \Delta(X,Y) = \sum_{i=1}^{n} \delta(x_i, y_i)$$

Equation 2-9 Distance Formula with Feature Weighting (left) and without (right)

A second type of feature weighting is Information Gain. Information Gain has been shown to be a popular feature weighting method, especially when used in conjunction with the IB1 algorithm (Daelemans et al. 1998a, 1998b, Daelemans 1999, Buchholz 1998a, Busser et al. 1999). Information Gain considers each individual feature and quantifies its relevance toward proper classification of the item. The weight of a given feature, *w_i*, is computed by calculating the difference in entropy, or uncertainty, between situations with and without knowledge of the feature (Equation 2-10).

$$w_i = H(C) - \sum_{v \in V_i} P(v) \times H(C \mid v)$$

Equation 2-10 Information Gain

A variation to the Information Gain method is the Gain Ratio algorithm. Information Gain tends to overestimate the relevance of features with large numbers of values. Features that possess many values have a very high Information Gain, but add no generalization to new instances (Daelemans et al. 2001). In an effort to normalize data, Quinlan (Quinlan 1993) has introduced the Gain Ratio formula as shown in Equation 2-11. The difference between the two methods is that the Gain Ratio formula is divided by a normalizing factor. The denominator is the entropy of the system, or 'split info'. The split info, or *si(i)*, helps to avoid a bias in favor of features with more values (van den Bosch 2000).

$$w_i = \frac{H(C) - \sum_{v \in V_i} P(v) \times H(C \mid v)}{si(i)}$$

where: $$si(i) = -\sum_{v \in V_i} P(v) \log_2 P(v)$$

Equation 2-11 Gain Ratio Formula

One of the disadvantages of applying the above methods is that the relevance of redundant features is often overestimated. A duplicated feature, for instance, results in an overestimation of the feature's weight by a factor of 2. This can dominate the similarity metric and thus lead to accuracy loss (Daelemans et al. 2001).

White and Liu (1994) have shown that Gain Ratio, despite the normalizing, still has an unwanted bias towards features with more values. Their reasoning is that the Gain Ratio statistic is not corrected for the number of degrees of freedom relating classes and values (Daelemans et al. 2001). White and Liu (1994) presented the Chi-squared statistic, which can be compared across conditions with different numbers of degrees of freedom. The Chi-squared formula (χ^2), is shown in Equation 2-12, where O_{ij} is the observed number of cases, and E_{ij} is the expected number of cases which should be in cell (v_i,c_j) in the contingency table (Daelemans et al. 2001).

$$\chi^2 = \sum_i \sum_j \frac{(E_{ij} - O_{ij})^2}{E_{ij}}$$

Equation 2-12 Chi-squared (χ^2)

The last of the feature weighting algorithms that will be discussed is the Shared Variance measure. Instead of using the χ^2 feature weighting, correction for the degrees of freedom can be done using Equation 2-13, where the denominator denotes the degrees of freedom. C is the number of classes and V is the number of values (Daelemans et al. 2001).

$$SV_i = \frac{\chi_i^2}{N \times \min(|C|,|V|) - 1}$$

Equation 2-13 Shared Variance

In this thesis, all five methods mentioned above are implemented. It will be shown later that recognition results depend significantly on which feature weighting method is utilized: Information Gain, Gain Ratio, Chi-squared, Shared Variance, or none at all.

2.7 *k*-values

IB1 is a nearest-neighbor (NN) approach, or a *k*-NN classifier. Once a similarity calculation is performed for each training instance, a ranking can be established of the closest match down to the lowest match. Instead of simply taking the classification of the single neighbor providing the highest match (i.e. $k = 1$), the most common classification of some number of highest scoring instances can be used, where *k* represents that number. Suppose some situation where the *k*-value is 7 and the top three scores yield a classification of tone #5, but the next four are classified as tone #2. The net result would be a classification of tone #2 because the most common classification within the top 7 nearest neighbors is used as the final result.

Increasing the number of nearest neighbors increases generalization of the system, but decreases efficiency (van den Bosch 1999b). Due to this computational inefficiency, some have tried methods to remove unnecessary instances from memory (Hart 1968, Gates 1972, Wilson 1972, Devijer and Kittler 1980), although that may be harmful (Daelemans et al. 1999). In this thesis, it was decided to use odd-numbered *k*-values from 1 to 15, with no removal of instances from memory. Only odd-numbered values were chosen in an attempt to limit the number of ties.

Without using *k*-values, classification must be assigned the tone associated with the highest scoring feature vector. It will be shown that using various values of *k* has a profound effect on recognition accuracy.

2.8 Class Voting Weights

If the *k*–values are very small (thus only a few nearest-neighbors), the final judgment can be unreliable and misleading. Larger values of *k* tend to offer higher accuracy, but if too high, mediocre matches will add unnecessary votes that may decrease overall accuracy. As a result, it is very important to find task-specific *k*-values that perform best. This thesis implements 4 different weighting functions for weighting the *k*-nearest neighbors: Normal, Inverse Linear, Inverse Distance, and Exponential Decay (Figure 2-2).

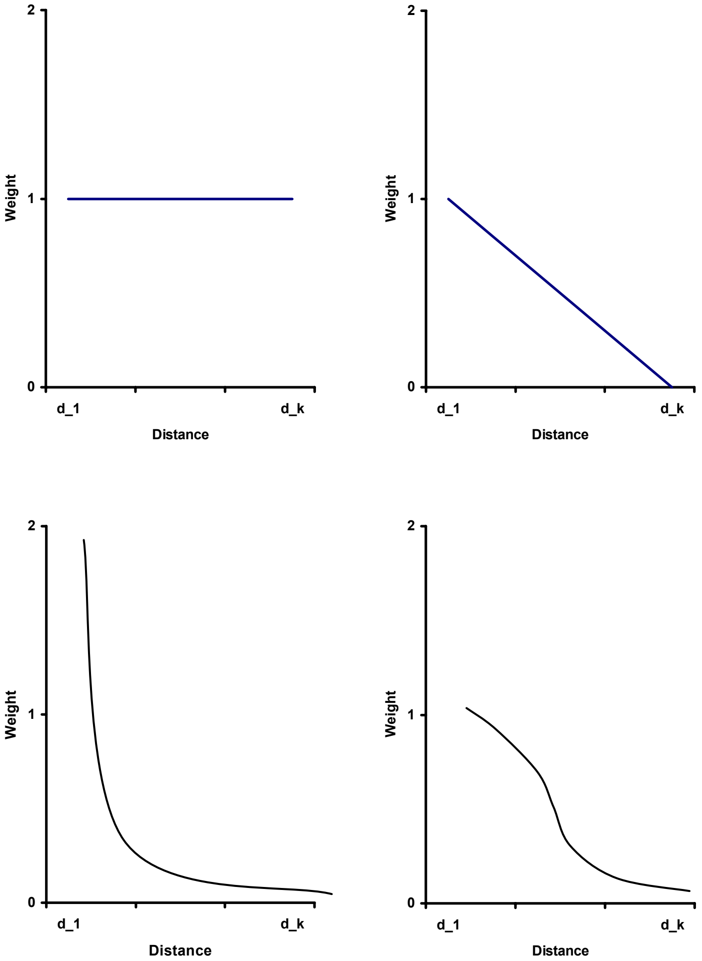


Figure 2-2 Class Voting Weight Functions: Normal (top left), Inverse Linear (top right), Inverse Distance (bottom left), and Exponential Decay (bottom right)

The most straightforward approach (i.e. 'Normal') is to equally weight all *k*-nearest neighbors. The end result is simply the classification with the highest number of votes. Although this approach is effortless and the least computationally demanding of the approaches, it does not perform as well as methods that weight the nearest neighbors' votes most heavily.

A second method is called Inverse Linear. The idea of weighting votes according to distance from the query was first proposed by Dudani (1976). Dudani's first proposal is a method where the nearest neighbor receives a weight of 1, the furthest a weight of 0, and the others receive a weight scaled linearly in between. This method of weighting is called Inverse Linear and is shown in Equation 2-14, where d_j is the distance to query, d_1 is the nearest neighbor, and d_k is the furthest neighbor.

$$w_j = \begin{cases} \dfrac{d_k - d_j}{d_k - d_1} & if\ d_k \neq d_1 \\ 1 & if\ d_k = d_1 \end{cases}$$

Equation 2-14 Inverse Linear Function

Another proposal by Dudani is the Inverse Distance function (Dudani 1976). The Inverse Distance equation is shown in Equation 2-15 where d_j is the distance to the query. Also, note that a small constant is usually added to the denominator to help avoid division by zero (Wettschereck, 1994).

$$w_j = \begin{cases} \dfrac{1}{d_j} & if \; d_j \neq 0 \end{cases}$$

Equation 2-15 Inverse Distance Function

The last function considered here is the Exponential Decay approach proposed by Shepard (1987). He argues that the relevance of a nearest-neighbor is an exponentially decreasing function of its distance from the query item. Equation 2-16 gives the Exponential Decay function, where α and β are constants determining slope and power of the function. Both α and β have been set to 1 for the experiments in this study.

$$w_j = e^{-\alpha d_j^{\beta}}$$

Equation 2-16 Exponential Decay Function

The four functions presented here attempt to determine which nearest-neighbor(s) should carry the most weight in the classification calculation. Experiments are performed in this thesis using all four methods to determine which function results in highest classification accuracy.

2.9 Classification Algorithm

The role of classification in this thesis is to decide which Cantonese tone best fits the data. The choice of a particular classification algorithm can have a profound effect upon results. The TiMBL software used in this thesis offers four separate classification algorithms to choose from: IB1, IGTREE, TRIBL, and IB2. Although similar, these algorithms vary in: 1) their definition of similarity, 2) the way the instances are stored in memory, and 3) the way the search through memory is conducted (Daelemans et al. 2001). A brief explanation of each is provided in Table 2-14.

ALGORITHM	DESCRIPTION
IB1	*k*-NN algorithm (default)
IGTREE	decision-tree-based optimization
TRIBL	hybrid of IB1 and IGTREE
IB2	incrementally edited machine-based learning

Table 2-14 Summary of TiMBL Algorithms

The default algorithm, IB1, was chosen for this work. The IB1 algorithm's primary distinction is that it provides maximal accuracy, yet at the expense of increased computational cost (Daelemans et al. 2001). If memory or computational power were an issue, one of the others could have been used. The result would be a faster, memory-friendly version but with a loss in accuracy. Test cases for this task confirmed that IB1 outperforms the other algorithms, albeit by a small margin.

We have just taken a look at the methodology used in this thesis. The next chapter will now present the results from using this process.

3 Results

This chapter will now present the results obtained using the methodology described in Chapter 2. This thesis presents results based on 4 main tone recognition systems developed through the course of this work: 6-tone/8-feature, 9-tone/8-feature, 6-tone/16-feature, and 9-tone/16-feature. Within each system, nearly every possible combination of algorithm, feature extraction, feature weighting, feature trimming, *k*-value, and class voting weights was tried in an attempt to find the best combination for each system.

With the exception of the normalization and algorithm results, results of individual speakers are not given. The results presented consist of the average of all 4 speakers' results. Because tens of thousands of experiments were performed, only the best score achieved is presented for each case. In many experiments, however, the average of all results is also provided to help draw conclusions.

3.1 Feature Extraction

One way to vary accuracy is to use different methods of feature extraction. Experiments were performed using both Auto Correlation and Cepstrum feature extraction methods. When calculating cepstrum features, a threshold (-*t*) can be set to help determine the voiced portion of data. Experiments were performed using –*t* values of 150 and 75, represented in tables below as CepA and CepB respectively. The value of 75 is the suggested default in relevant documentation; the value of 150 was chosen because it captured more data by raising the threshold in detecting voiced portions in speech. In 16-feature experiments, CepA and CepB were further split into CepA/B(s), CepA/B(0) and CepA/B(100) to indicate feature stretching, zero-addition, and 100-addition respectively (see Chapter 2.2.4).

3.1.1 Speaker-Dependent Results

It appears that the Auto Correlation method (AC) is the one best suited for 8-feature experiments. Table 3-1 shows that Auto Correlation produced the best results as well as the best average in 6-tone/8-feature experiments. Although CepB does have a higher "best result" in 9-tone/8-feature experiments, the average accuracy for Auto Correlation was noticeably higher, indicating that it is generally better.

	6-Tone		9-Tone	
	Best	Average	Best	Average
AC	90.5	86.8	80.6	75.6
CepA	88.6	83.2	78.0	69.5
CepB	89.8	85.9	80.9	74.5

Table 3-1 Feature Extraction Results (Speaker-Dependent; 8-feature)

It is less clear which method works best in 16-feature experiments. In 6-tone/16-feature experiments, CepstrumB using either zero-addition (CepB(0)) or 100-addition (CepB(100)) attained best results (Table 3-2).

	6-Tone		9-Tone	
	Best	Average	Best	Average
AC	90.4	87.8	82.5	77.7
CepA(s)	89.5	86.4	77.6	72.9
CepA(0)	89.5	86.1	86.6	82.0
CepA(100)	90.1	86.6	87.0	82.2
CepB(s)	89.5	87.0	81.3	76.4
CepB(0)	90.9	87.3	86.6	81.9
CepB(100)	90.5	87.6	86.5	81.7

Table 3-2 Feature Extraction Results (Speaker-Dependent; 16-feature)

In 9-tone/16-feature experiments, best results were achieved using the zero-addition and 100-addition methods of CepstrumA and CepstrumB. It does appear, however, that CepstrumA(100) is best in this case. Not only was the best result obtained using this method, but overall average also indicated it is the best approach for 9-tone/16-feature recognition.

3.1.2 Speaker-Independent Results

Recognition results for speaker-independent experiments are less diverse due to a much larger test set. As a result, even marginal improvements can prove significant. Results are similar for all three methods of feature extraction in 6-tone/8-feature experiments (Table 3-3). CepstrumA does, however, seem to be the best approach. In 9-tone/8-feature experiments, however, it is clear that Auto Correlation is the best approach since recognition accuracy for this method outperformed the others by a strong margin.

	6-Tone		9-Tone	
	Best	Average	Best	Average
AC	80.4	79.0	73.4	71.2
CepA	80.9	79.5	72.0	69.7
CepB	80.6	79.2	70.8	69.2

Table 3-3 Feature Extraction Results (Speaker-Independent; 8-feature)

Table 3-4 shows results for 6-tone/16-feature experiments. Again, as in the 6-tone/8-feature results, it is rather uncertain which method is best; no method stands out as significantly better than the others. In 9-tone/16-feature experiments, CepstrumA(0) was the clear winner. Not only were the best results obtained using this process, but the overall average is also much higher than the others.

	6-Tone		9-Tone	
	Best	Average	Best	Average
AC	81.8	80.6	74.8	73.4
CepA(s)	81.0	80.4	71.9	70.3
CepA(0)	81.7	81.1	77.5	76.8
CepB(s)	81.2	80.6	72.2	71.0
CepB(0)	81.4	80.7	76.7	75.8

Table 3-4 Feature Extraction Results (Speaker-Independent; 16-feature)

Auto Correlation appears to be the best method for 8-feature experiments. In 16-feature experiments, however, the Cepstrum approaches usually outperformed Auto Correlation. CepstrumA is probably the best choice for 16-feature experiments, as CepstrumB does not seem to be particularly useful in Speaker-Independent experiments.

Overall, it is not very clear which feature extraction approach is best suited for tone recognition, as results vary for each system. It is apparent, however, that for any given experiment the method chosen can have a considerable impact on recognition accuracy.

3.2 Normalization

Normalization of data is only performed for speaker-independent experiments. The following reports on an experiment to illustrate the positive effects of normalizing data. Table 3-5 compares recognition results from normalized data to results from data not normalized for each of the four speakers.

	6-Tone		9-Tone	
	Not Normalized	Normalized	Not Normalized	Normalized
cs01f	28.2	85.5	27.3	74.5
cs02m	71.8	89.3	66.1	78.4
cs03f	37.8	67.9	30.8	62.3
cs04m	67.9	81.4	62.9	74.8
Average	51.4	81.0	46.8	72.5

Table 3-5 Normalization Results (8-features)

In 6-tone/8-feature experiments, normalizing data increased average accuracy from 51.4% to 81.0%. Likewise, normalizing data improved average recognition rate from 46.8% to 72.5% for 9-tone/8-feature experiments.

As we will see again later, tone data from speaker cs03f is quite different from the other speakers'. Even after normalization of data, recognition results for cs03f (67.9% in 6-tone; 62.3% in 9-tone) are very poor in comparison to other speakers. Manual inspection of data from cs03f indicates that the speaker is indeed native, but her tones do not follow "standard" contours.

3.3 Feature Trimming

3.3.1 Speaker-Dependent Results

One of the techniques mentioned earlier (Chapter 2.2.6) is feature trimming, a process where certain features are ignored during the recognition process in an attempt to increase recognition performance. In the 6-tone/8-feature system, 90.5% accuracy was achieved without implementing feature trimming (Table 3-6). The best accuracy with feature trimming (89.3%) was achieved when the first and last features (i.e. I1-8) were ignored. Reducing the first and last pair of features (i.e. I1-2,7-8) further reduced accuracy to 86.4%. Examining averages of results yields a similar trend — the more features that are ignored, the more recognition accuracy decreases. Examination of results gathered from the 9-tone/8-feature system exhibit a similar trend. Highest results are achieved if all available information is used.

	6-Tone		9-Tone	
Method	Best	Average	Best	Average
No Trimming	90.5	87.9	80.9	78.1
I1-8	89.3	85.8	78.6	73.5
I1-2,7-8	86.4	82.2	74.5	67.9

Table 3-6 Feature Trimming Results (Speaker-Dependent; 8-feature)

Table 3-7 shows results for the 6-tone/16-feature and 9-tone/16-feature systems. Because 16-feature experiments have more features than 8-feature experiments, more variations of feature trimming were attempted. Similar to the 8-feature experiments, 16-feature experiments exhibit a steady decrease in performance when features on both ends are ignored (i.e. I-16; I1-2,15-16; I1-3,14-16; I1-4,13-16 in Table 3-7). In these instances, accuracy continues to decrease as more features are ignored. This is in direct contradiction to a common practice used in tone recognition (see Chapter 2.2.6).

	6-Tone		9-Tone	
Method	Best	Average	Best	Average
No Trimming	90.0	87.8	86.8	81.4
I1,16	89.8	87.3	86.0	80.3
I1-2,15-16	89.3	86.7	83.9	78.6
I1-3,14-16	88.3	85.4	81.0	76.1
I1-4,13-16	86.3	83.4	76.8	72.4
I1	90.4	88.2	86.8	81.7
I1-2	90.9	88.4	87.0	81.9
I1-3	90.4	88.4	86.0	81.7

Table 3-7 Feature Trimming Results (Speaker-Dependent; 16-feature)

Results are encouraging, however, when only features at the beginning of a feature vector are ignored. In 6-tone/16-feature experiments, accuracy is improved when the first 1, 2, or 3 features are ignored (Table 3-7). The best result improved from 90.0% to 90.9% when the first 2 features were ignored.

9-tone/16-feature experiments show similar results ignoring the first 2 features yielding the best results. The benefit, however, seems less drastic in 9-tone experiments. Ignoring the first 2 features increases accuracy only from 86.8% to 87.0% — an increase of only 0.2% compared to the 0.9% increase observed in the 6-tone experiments.

3.3.2 Speaker-Independent Results

Feature trimming was not attempted in either the 6-tone or 9-tone 8-feature systems as previous experiments strongly suggested this would only harm recognition accuracy. In 16-feature experiments, attempts were limited to eliminating the first 1, 2, or 3 features as these exhibited best results in previous experiments. In 6-tone/16-feature experiments very little change is noted in best scores obtained and absolutely no change in average word accuracy (Table 3-8). The results for 9-tone/16-feature experiments were similar. It does appear, however, that ignoring more features tended to slightly decrease average accuracy.

	6-Tone		9-Tone	
Method	Best	Average	Best	Average
No Trimming	81.8	80.7	77.3	73.5
I1	81.6	80.7	77.4	73.6
I1-2	81.7	80.7	77.5	73.4
I1-3	81.6	80.7	77.3	73.2

Table 3-8 Feature Trimming Results (Speaker-Independent; 16-feature)

In sum, ignoring any features in an 8-feature system results in a loss of accuracy. When more features are used (such as in the 16-feature systems), however, ignoring the first two features produces the best results. This is probably a result of greater tone variation in the beginning portions of tone data. In the speaker-independent systems, no noticeable change in results was observed by ignoring features.

3.4 Feature Weighting

3.4.1 Speaker-Dependent Results

In 6-tone/8-feature experiments, Gain Ratio and Information Gain produced the best results with an accuracy of 90.5% and 90.4% achieved respectively (Table 3-9). The average recognition result was almost equal for each method except the Chi-squared approach. In this case, average recognition results for Chi-squared was nearly a full percentage point lower than the other methods.

	6-Tone		9-Tone	
Method	Best	Average	Best	Average
None	89.4	85.3	80.5	74.3
Gain Ratio	90.5	85.4	80.9	75.0
Information Gain	90.4	85.5	80.8	74.9
Chi-squared	89.9	84.7	80.8	74.3
Shared Variance	89.9	85.5	80.8	74.9

Table 3-9 Feature Weighting Results (Speaker-Dependent; 8-feature)

Not using a method of feature weighting resulted in lower recognition rate in 9-tone/8-feature experiments. The other approaches all exhibited similar results except that the average recognition rate using Chi-squared was slightly lower than the others.

A recognition rate of 90.9% was achieved using both Chi-squared and Shared Variance feature weighting methods (Table 3-10). Again, the average result using Chi-squared was noticeably lower than results from the other methods.

	6-Tone		9-Tone	
Method	Best	Average	Best	Average
None	90.5	87.0	86.8	79.4
Gain Ratio	90.4	87.1	87.0	79.5
Information Gain	90.4	87.2	86.8	79.6
Chi-squared	90.9	86.2	86.8	78.2
Shared Variance	90.9	87.3	86.8	79.6

Table 3-10 Feature Weighting Results (Speaker-Dependent; 16-feature)

Best results achieved were nearly the same for all methods in 9-tone/16-feature experiments. Using Gain Ratio produced the highest score of 87.0%. Once again, the average results obtained using the Chi-squared method was quite low in comparison with the others.

3.4.2 Speaker-Independent Results

In both 6-tone/8-feature and 9-tone/8-feature experiments, no particular feature weighting method separated itself as the best approach (Table 3-11). It is clear, however, that not using a method produced lower recognition accuracy in the 9-tone/8-feature experiments. Similar to the speaker-dependent results, the Chi-squared method achieved the lowest scores in total average.

	6-Tone		9-Tone	
Method	Best	Average	Best	Average
None	80.7	79.3	72.2	70.1
Gain Ratio	80.8	79.6	73.4	70.6
Information Gain	80.5	79.5	73.3	70.5
Chi-squared	80.9	78.2	73.0	68.6
Shared Variance	80.9	79.5	73.0	70.4

Table 3-11 Feature Weighting Results (Speaker-Independent; 8-feature)

Each method achieved nearly identical scores in both the 6-tone/16-feature and 9-tone/16-feature experiments (Table 3-12).

	6-Tone		9-Tone	
Method	Best	Average	Best	Average
Gain Ratio	81.8	80.7	77.5	73.4
Information Gain	81.6	80.7	77.4	73.4
Chi-squared	81.8	80.7	77.4	73.5
Shared Variance	81.8	80.7	77.4	73.5

Table 3-12 Feature Weighting Results (Speaker-Independent; 16-feature)

In sum, there is no clear best method of feature weighting. The Gain Ratio and Information Gain methods are very similar and the above results show that between the two of these approaches, Gain Ratio is best suited for this task. Information Gain never achieved better results than Gain Ratio. Chi-squared and Shared Variance are also very similar in their approach, but in this case, the best results achieved are identical in every table above. Average scores are much higher for Shared Variance, however, so this might be a better choice. With only a few exceptions, the Gain Ratio method achieved the highest accuracy in each case. It would seem that this approach is the one best suited for the Cantonese tone recognition task. It is clear that in systems using 9 tones and 8 features, not weighting the features harms accuracy. It is unclear, however, why the average score using the Chi-squared methodology in every system was noticeably lower than the others. Its lack of a normalizing element may be a key contributor to its relative inefficiency.

3.5 *k*-values

3.5.1 Speaker-Dependent Results

Changing values of *k* can have a strong effect on recognition rate. In 6-tone/8-feature experiments, the overall result is that the higher the *k*-value, the higher the accuracy (Table 3-13). While the average continued to improve with an increase of *k*, the absolute best score was obtained using a *k*-value of 13. Best results for 9-tone/8-feature experiments were obtained using *k*-values of 11 or 13. Using a *k*-value of 9, however, achieved the best results when averaging all experiments.

	6-Tone		9-Tone	
k-value	Best	Average	Best	Average
1	86.8	82.8	78.8	74.9
3	89.3	84.3	80.0	77.4
5	89.8	85.2	80.4	78.0
7	89.8	85.7	80.1	78.2
9	90.4	86.0	80.3	78.4
11	90.4	86.1	80.6	78.2
13	90.5	86.2	80.6	78.1
15	90.1	86.2	80.4	78.2

Table 3-13 *k*-value Results (Speaker-Dependent; 8-feature)

In 6-tone/16-feature experiments, average recognition rate continually improved with an increase of *k*-value (Table 3-14). The highest singular result, however, was achieved using a *k*-value of 5. There was no clear trend in 9-tone/16-feature results. In this case, using a *k*-value of 11 obtained the highest result, whereas the highest average was achieved by using 7 as the *k*-value.

	6-Tone		9-Tone	
k-value	Best	Average	Best	Average
1	88.3	85.0	84.0	77.6
3	90.3	86.5	86.1	79.0
5	90.9	87.2	86.8	79.6
7	90.4	87.3	86.6	79.8
9	90.5	87.4	86.5	79.7
11	90.3	87.4	87.0	79.6
13	90.5	87.4	86.6	79.5
15	90.4	87.4	86.8	79.3

Table 3-14 *k*-Value Results (Speaker-Dependent; 16-feature)

3.5.2 Speaker-Independent Results

Due to the larger test size for speaker-independent experiments, the trend is much more apparent than in the speaker-dependent systems (Table 3-15). Each of these experiments illustrates that the higher the *k*-value, the higher the average recognition accuracy. Furthermore, the highest score obtained in each experiment was 15. The lone exception was in the 9-tone/16-feature experiment where a *k*-value of 13 resulted in the best score.

	6-Tone		9-Tone	
k-value	Best	Average	Best	Average
1	77.9	77.5	68.8	67.0
3	79.6	78.5	71.6	68.9
5	79.9	79.1	72.5	69.9
7	80.2	79.4	73.1	70.6
9	80.6	79.6	73.1	70.9
11	80.8	79.8	73.3	71.0
13	80.8	79.9	73.2	71.0
15	80.9	79.9	73.4	71.1

Table 3-15 *k*-Value Results (Speaker-Independent; 8-feature)

	6-Tone		9-Tone	
k-value	Best	Average	Best	Average
5	81.2	79.9	73.9	69.9
7	81.4	80.4	74.3	71.1
9	81.5	80.6	74.7	71.7
11	81.6	80.8	74.7	72.0
13	81.8	80.8	74.8	72.3
15	81.8	80.9	74.6	72.4

Table 3-16 *k*-Value Results (Speaker-Independent; 16-feature)

3.6 Class Voting Weights

3.6.1 Speaker-Dependent Results

On average, using the normal weighting method performed best for 6-tone/8-feature experiments (Table 3-17). The best overall performance, however, was achieved using the Inverse Linear approach, in which 90.5% accuracy was attained. In the 9-tone/8-feature experiments, Inverse Distance method achieved the best results. The Inverse Distance approach also achieved the highest average score (along with the Inverse Linear method).

	6-Tone		9-Tone	
Method	Best	Average	Best	Average
Normal	90.1	85.6	80.6	73.2
Exponential Decay	90.4	85.0	80.8	72.9
Inverse Linear	90.5	85.5	80.8	73.4
Inverse Distance	90.0	85.1	80.9	73.4

Table 3-17 Class Voting Weights Results (Speaker-Dependent; 8-feature)

Results for both the 6-tone/16-feature experiments as well as the 9-tone/16-feature experiments mirrored the results from the 8-feature experiments presented above (Table

3-18). Results from the 6-tone/16-feature experiments show the same trend as the 6-tone/8-feature experiments in that Inverse Linear produced the best overall performance, whereas using the normal weighting method attained the best average score.

	6-Tone		9-Tone	
Method	Best	Average	Best	Average
Normal	90.5	86.9	86.3	79.6
Exponential Decay	90.8	85.0	86.8	77.6
Inverse Linear	90.9	86.5	86.6	79.0
Inverse Distance	90.4	87.2	87.0	79.6

Table 3-18 Class Voting Weights Results (Speaker-Dependent; 16-feature)

The 9-tone/16-feature results show the same trend as the 9-tone/8-feature results in that the Inverse Distance method achieves top scores for both best overall result as well as best average score.

3.6.2 Speaker-Independent Results

Results from 6-tone/8-feature experiments show that using the Inverse Linear approach produced the best recognition rate (Table 3-19). The Inverse Linear and Normal weighting methods achieved the best average result. In 9-tone/8-feature experiments, the Inverse Distance approach got the highest score. When averaging all experiments, Inverse Linear produced the highest score of 70.7%.

	6-Tone		9-Tone	
Method	Best	Average	Best	Average
Normal	80.7	79.6	72.8	70.5
Exponential Decay	80.7	78.6	73.2	69.2
Inverse Linear	80.9	79.6	73.3	70.7
Inverse Distance	80.6	79.1	73.4	69.8

Table 3-19 Class Voting Weights Results (Speaker-Independent; 8-feature)

Both the 6-tone/16-feature and 9-tone/16-feature experiments show similar results (Table 3-20). In both experiments, using the Inverse Linear method produced the best results, achieving an accuracy of 81.8% and 77.5% in the 8-feature and 16-feature experiments respectively. In both experiments, however, using Inverse Distance produced the highest average.

	6-Tone		9-Tone	
Method	Best	Average	Best	Average
Inverse Linear	81.8	80.1	77.5	72.4
Inverse Distance	81.6	80.6	77.3	73.2

Table 3-20 Class Voting Weights Results (Speaker-Independent; 16-feature)

In sum, it appears that for speaker-dependent systems, using the Inverse Distance approach works best for the 9-tone experiments, whereas the Inverse Linear method achieves best results for the 6-tone experiments. In the speaker-independent experiments, using the Inverse Linear approach produces the highest scores in each case except for 9-tone/8-feature experiments in which it is a close second (Table 3-19).

3.7 Classification Algorithms

The following reports work done to test whether IB1 is indeed the classification algorithm best suited for the tone recognition task. As noted in Chapter 2, IB1 is the default and usually most accurate method, though at the expense of high computational cost.

Recognition results for speaker cs01f are provided in Table 3-21. In preliminary tests, features were extracted using the Auto Correlation method, the *k*-value was set to 1, and no features were trimmed. Because the *k*-value was set to 1, there was no need for class voting weights.

	6-Tone		9-Tone	
Experiment	IGTREE	IB1	IGTREE	IB1
MVDM (no weighting)	80.0	89.0	63.5	78.5
MVDM (gain ratio)	80.0	89.5	63.5	79.0
MVDM (information gain)	80.0	91.0	63.5	78.5
MVDM (chi-squared)	80.0	89.0	63.5	79.0
MVDM (shared variance)	80.0	89.0	63.5	79.0
Numeric (no weighting)	80.0	91.0	63.5	81.5
Numeric (gain ratio)	80.0	90.5	63.5	80.5
Numeric (information gain)	80.0	90.0	63.5	79.5
Numeric (chi-squared)	80.0	93.0	63.5	83.5
Numeric (shared variance)	80.0	93.0	63.5	83.5
AVERAGE	80.0	90.5	63.5	80.3

Table 3-21 Algorithm Results (8-feature)

In 6-tone/8-feature experiments, average results increased from 80.0% to 90.5% when using the IB1 classification algorithm. In 9-tone/8-feature experiments, average

IGTREE accuracy of 63.5% was improved to 80.3% when using IB1. This study shows that IB1 is indeed more appropriate for the Cantonese tone recognition task.

3.8 Number of Features

The decision of how many features to extract from a file is very important. Clearly, the more features that are used, the more data is present to aid the training and classification processes. As expected, using 16 features (84.3%) produced a higher recognition rate than using 8 features (81.4%)(Table 3-22).

	8 Features	16 Features
Speaker-Dependent 6-Tone	90.5	90.9
Speaker-Dependent 9-Tone	80.9	87.0
Speaker-Independent 6-Tone	80.9	81.8
Speaker-Independent 9-Tone	73.4	77.5
AVERAGE	81.4	84.3

Table 3-22 Results for 8-feature vs. 16-feature comparison

Using 16 features instead of 8 produced a relatively small increase in accuracy in 6-tone systems. A 0.4% increase was noted for speaker-dependent 6-tone systems, as well as a 0.9% increase for speaker-independent 6-tone systems.

Using 16 features greatly enhanced recognition rate for 9-tone systems however. For speaker-dependent 9-tone experiments, the best score was improved by 6.1%. The speaker-independent equivalent was improved by 4.1%.

It is clear that using 16 features produces better accuracy than using 8 features. The benefit is far greater for 9-tone recognition systems. The optimal number of features to use for the Cantonese tone recognition task is still not clear. Two sample experiments were performed using 32 features, for example, and accuracy decreased from that obtained using 16 features. This shows that using more features does not necessarily yield higher accuracy.

The optimal number of features to use may be a function of the average number of features returned from the feature extraction method. In cases where the number of features is large, such as 32, accuracy decreases because a large number of test cases do not have the required number of features, thus necessitating data creation through feature manipulation steps mentioned earlier in Chapter 2.2.

As noted earlier, one benefit of using an 8-feature system rather than a 16-feature system is less computational cost. Although this may be a viable option for 6-tone systems, choosing to use 8 features for a 9-tone system considerably decreases performance.

3.9 Number of Tones

Table 3-23 shows that simplifying the Cantonese tone recognition task to only 6 tones greatly increases recognition rate. Average accuracy for 9-tone systems is 79.7%, whereas the average for 6-tone systems is improved to 86.0%. The difference is greater for systems using only 8 features. Speaker-dependent 8-feature accuracy was improved by 9.6% by simplifying the task to a 6-tone system. Likewise, the speaker-independent 8-feature system exhibited a 7.5% increase. It is possible that 9-tone systems require a larger number of features than 6-tone systems do. This would account for the relatively low results for the 9-tone, 8-feature experiments.

	9-Tone	6-Tone	Increase(%)
Speaker-Dependent 8-Features	80.9	90.5	+9.6
Speaker-Dependent 16-Features	87.0	90.9	+3.9
Speaker-Independent 8-Features	73.4	80.9	+7.5
Speaker-Independent 16-Features	77.5	81.8	+4.3
AVERAGE	79.7	86.0	+6.3

Table 3-23 Results for 6-tone vs. 9-tone comparison

For experiments using 16 features, the change is less dramatic. An increase of 3.9% is noted for the speaker-dependent 16-feature system, as well as a 4.3% increase for the speaker-independent system.

It is clear that if a recognition task can be done using a 6-tone classification, recognition accuracy will be higher than by using a 9-tone classification system.

3.10 Speaker Dependency

According to Table 3-24, speaker-dependent systems outperformed their speaker-independent counterparts. The average result for speaker-dependent systems is 87.3%, compared to an average of 78.4% for speaker-independent systems.

	Speaker Dependent	Speaker Independent
6-Tone, 8-Features	90.5	80.9
6-Tone, 16-Features	90.9	81.8
9-Tone, 8-Features	80.9	73.4
9-Tone, 16-Features	87.0	77.5
AVERAGE	87.3	78.4

Table 3-24 Results for Speaker-Dependent and Speaker-Independent systems

Remember, however, that the speaker-independent results presented here are not truly speaker-independent per se. In speaker-independent experiments, average tonal frequency was calculated for use in the normalization process. Of course, a speaker's average frequency must be known a priori in order for the results presented here to be accurate. One option for implementation would be to perform a minimal amount of speaker training. Collecting just a few speech samples would probably be suitable. Another option is to perform online training (i.e. continually recalculate average frequency). In this case, the first few speech items may be normalized poorly, but accuracy would quickly improve.

3.11 Best Methods

Table 3-25 shows the best combination of parameters found for each system. For example, in speaker-dependent 6-tone/8-feature experiments the highest result of 90.5% was achieved while using the Auto Correlation method of extracting features, no feature trimming, Gain Ratio approach for feature weighting, a *k*-value of 13, and Inverse Distance as the class voting weighting method. A legend of abbreviations used in Table 3-25 is provided in Table 3-26 for the reader's convenience. Chapter 2 contains additional information regarding the items presented here.

Table 3-25 does not present any new information as earlier sections have already covered the data in more detail. The data is presented here as a concise summary of results for all 8 systems.

	FE	Trim	FW	*k*	CVW	%
Speaker-Dependent 6-Tone/8-Feature	AC	---	gr	13	ID	90.5
Speaker-Dependent 9-Tone/8-Feature	CepB	---	gr	11/13	IL	80.9
Speaker-Dependent 6-Tone/16-Feature	CepB(0)	I1-2	sv/χ^2	5	ID	90.9
Speaker-Dependent 9-Tone/16-Feature	CepA(100)	I1-2	gr	11	IL	87.0
Speaker-Independent 6-Tone/8-Feature	CepA	---	sv	15	ID	80.9
Speaker-Independent 9-Tone/8-Feature	AC	---	gr	15	IL	73.4
Speaker-Independent 6-Tone/16-Feature	CepA(s)	---	sv/χ^2	13	ID	81.8
Speaker-Independent 9-Tone/16-Feature	CepA(0)	I1-2	gr	15	ID	77.5

Table 3-25 Parameters producing best results for all 8 systems

FE	Feature Extraction Method
Trim.	Features Trimmed
FW	Feature Weighting Method
CVW	Class Voting Weighting Method
AC	Auto Correlation
CepB	CepstrumB
CepB(0)	CepstrumB(Zero-Addition)
CepA(100)	CepstrumA(100-Addition)
CepA	CepstrumA
CepA(s)	CepstrumA(Stretched)
CepA(0)	CepstrumA(Zero-Addition)
I1-2	Ignore 1st and 2nd feature
gr	Gain Ratio
sv	Shared Variance
χ^2	Chi-squared
ID	Inverse Distance
IL	Inverse Linear

Table 3-26 Legend of Abbreviations for Table 3-25

3.12 Error Analysis

A common tool for analyzing classification errors is a confusion matrix, where classification errors can be analyzed. Confusion matrices are presented below to facilitate the analysis of errors produced during tone classification. Numbers in the matrices represent sums of results obtained from all four speakers. More detail for each individual speaker is provided in Appendices A-H. In each system, data is only presented for the method producing the highest accuracy. For more detail on which set of parameters produced these results, refer back to Chapter 3.11.

3.12.1 Speaker-Dependent Results

Interestingly, Table 3-27 shows that tone #5 is the most difficult to recognize, only being recognized correctly 80.0% of the time. No mention of this phenomenon is found in previous literature. Recognition of tones #3 and #6 is also relatively poor achieving 87.2% and 86.8% accuracy respectively. In contrast, tones #1, #2, and #4 all perform very well.

	1	2	3	4	5	6	%
1	155	0	5	1	0	3	94.5%
2	0	147	1	0	8	0	94.2%
3	3	0	150	0	0	19	87.2%
4	3	0	0	99	1	1	95.2%
5	0	3	3	3	48	3	80.0%
6	0	0	16	3	0	125	86.8%
						AVG	90.5%

Table 3-27 Confusion Matrix for Best Method (Speaker-Dependent; 6-tone/8-feature)

The majority of errors appear to be due to a misclassification between tones with similar contours. Tones #1, #3, and #6 all have identical contour patterns and are primarily distinguished by frequency only. Misclassifications between the level tones (i.e. #1, #3, and #6) account for 46 of the 76 errors reported (61%). Confusion between tones #3 and #6 is much greater than the confusion of either tone with tone #1. This is very understandable as the average frequency between the two tones is quite close.

In the 9-tone/8-feature system, similar trends were noted (Table 3-28). The majority of misclassifications are again most common between tones with identical contours. The short-duration tones (i.e. tones #7, #8, and #9) had by far the lowest recognition rates. 67 of the 153, or 44%, of the errors are due to the inability to distinguish between tones differing by length only (i.e. tone #1 with #7, #3 with #8, and #6 with #9). Another difficulty is in distinguishing between tones #2 and #5. This problem accounts for 15 of the errors. Very high accuracy continues to be achieved for tones #1, #2, and #4.

	1	2	3	4	5	6	7	8	9	%
1	111	0	3	0	0	0	2	0	0	95.7%
2	0	145	0	1	7	1	0	0	2	92.9%
3	3	0	107	0	0	12	0	4	2	83.6%
4	0	0	0	100	0	4	0	0	0	96.2%
5	0	8	1	1	46	4	0	0	0	76.7%
6	0	0	9	1	2	48	0	0	4	75.0%
7	15	0	0	0	0	0	33	0	0	68.8%
8	0	1	16	1	2	2	0	16	6	36.4%
9	0	0	4	2	1	26	0	6	41	51.3%
									AVG	80.9%

Table 3-28 Confusion Matrix for Best Method (Speaker-Dependent; 9-tone/8-feature)

The conclusions drawn from the 6-tone/8-feature experiments are the same as the 6-tone/16-feature experiments shown in Table 3-29. Tone #5 achieves the worst accuracy. Recognition for tones #1, #2, and #4 are again very high. Distinguishing between tones #2 and #5, as well as distinguishing between #3 and #6 continues to be extremely difficult, and a common source of errors for this recognizer.

	1	2	3	4	5	6	%
1	161	0	3	0	0	0	98.2%
2	0	146	1	1	6	2	93.6%
3	3	0	143	0	1	21	85.1%
4	0	0	0	103	0	1	99.0%
5	0	8	1	1	46	4	76.7%
6	0	0	17	2	1	128	86.5%
						AVG	90.9%

Table 3-29 Confusion Matrix for Best Method (Speaker-Dependent; 6-tone/16-feature)

Table 3-30 shows the confusion matrix for the 9-tone/16-feature system. Accuracy is high for tones #1 and #4, but results for tone #2 are lower in comparison to earlier findings. Results are low for tone #5. Previously, recognition rate for tones #7, #8, and #9 were much lower than recognition rate for other tones. Although they still perform poorly, the difference is not as great. In fact, recognition accuracy for tone #7 performed extremely well, even better than many of the other tones. Interestingly, tone #3 and #6 continue to be confused, as well as tone #2 with tone #5.

	1	2	3	4	5	6	7	8	9	%
1	109	0	3	0	0	0	4	0	0	94.0%
2	0	138	0	1	10	1	0	0	6	88.5%
3	1	0	111	0	0	12	0	0	0	89.5%
4	0	0	0	103	0	1	0	0	0	99.0%
5	0	10	1	1	45	3	0	0	0	75.0%
6	0	0	7	1	0	55	0	0	1	85.9%
7	5	0	1	0	0	0	42	0	0	87.5%
8	0	0	4	0	0	1	0	31	8	70.5%
9	0	0	1	5	0	11	0	5	62	73.8%
									AVG	87.0%

Table 3-30 Confusion Matrix for Best Method (Speaker-Dependent; 9-tone/16-feature)

3.12.2 Speaker-Independent Results

In the speaker-independent 6-tone/8-feature system, tone #1 is recognized extremely well at 96.1% (Table 3-31). Tone #5 is recognized poorly at only 61.0%. 572 of the 1369 errors (42%) are due to the inability to distinguish the level tones (i.e. tones #1, #3, and #6) from each other. Worthy of note, it appears that many tones were misrecognized as tone #6. This is particularly surprising because this behavior was not exhibited in the speaker-dependent system.

	1	2	3	4	5	6	%
1	1622	10	52	3	0	1	96.1%
2	3	1000	17	9	144	91	79.1%
3	61	3	1079	1	1	251	77.3%
4	0	0	0	796	2	190	80.6%
5	0	78	11	11	332	112	61.0%
6	0	6	207	91	16	968	75.2%
						AVG	80.9%

Table 3-31 Confusion Matrix for Best Method (Speaker-Independent; 6-tone/8-feature)

Recognition errors for tones #7, #8, and #9 are the primary source of errors for the 9-tone/8-feature system (Table 3-32). Tone #3 is also not recognized very well. Accuracy is relatively high, however, for tones #1 and #2. The vast majority of errors continue to be a result of confusion between level tones (i.e. #1, #3, #6, #7, #8, #9).

	1	2	3	4	5	6	7	8	9	%
1	1112	0	139	9	0	11	35	1	1	85.0%
2	1	1160	1	2	96	3	0	1	0	91.8%
3	116	0	637	0	0	176	4	55	24	62.9%
4	59	1	0	768	27	59	26	0	48	77.7%
5	1	36	16	6	421	51	0	4	9	77.4%
6	1	0	126	11	15	654	0	23	34	75.7%
7	149	1	20	7	0	1	184	18	0	48.4%
8	20	1	119	0	7	41	13	136	46	35.5%
9	1	1	47	20	5	109	0	51	190	44.8%
									AVG	73.4%

Table 3-32 Confusion Matrix for Best Method (Speaker-Independent; 9-tone/8-feature)

Tones #1 and #2 perform very well in the 6-tone/16-feature system achieving recognition rates of 89.4% and 92.7% respectively (Table 3-33). 782 of the 1304 errors (60%) were due to confusion between the level tones. Similar to the speaker-independent 6-tone/8-feature results, many tones are misrecognized as tone #6. It is unclear why this is only the case for speaker-independent 6-tone systems.

	1	2	3	4	5	6	%
1	1509	1	152	15	0	11	89.4%
2	2	1172	5	2	80	3	92.7%
3	137	1	985	1	4	267	70.6%
4	116	0	0	745	22	105	75.4%
5	2	28	26	5	421	62	77.4%
6	3	3	212	27	10	1033	80.2%
						AVG	81.8%

Table 3-33 Confusion Matrix for Best Method (Speaker-Independent; 6-tone/16-feature)

Table 3-34 shows the confusion matrix for the 9-tone/16-feature system. Best recognition rate is for tone #1 achieving 94.5% recognition accuracy. Interestingly, the worst recognition is for tone #8 with only 49.2% accurately classified.

	1	2	3	4	5	6	7	8	9	%
1	1236	3	35	3	0	0	29	2	0	94.5%
2	0	954	22	7	172	66	12	3	28	75.5%
3	46	0	784	0	0	165	0	9	4	77.8%
4	0	0	0	826	1	104	0	0	57	83.6%
5	0	53	11	4	377	97	0	1	1	69.3%
6	0	0	178	35	10	612	0	5	24	70.8%
7	53	6	4	0	0	0	300	15	2	78.9%
8	9	2	88	1	0	10	9	189	76	49.2%
9	0	3	14	44	1	61	0	26	275	64.9%
									AVG	77.5%

Table 3-34 Confusion Matrix for Best Method (Speaker-Independent; 9-tone/16-feature)

It is clear that the primary weakness of this approach is its inability to distinguish between tones with identical contours. In all systems analyzed, tone #1 is by far the most easily recognized tone. This is surprising since the tonal contour is identical to many of the other tones, but accuracy remains high probably due to its higher frequency, thus clearly separating itself from the other tones. Because the tonal contours for tones #2 and #5 are similar, as well, it is interesting to note that recognition was also found to have problems distinguishing these two tones. Surprisingly, tone #5 was particularly difficult to recognize.

In 6-tone systems, the majority of errors result from confusion between tones #1, #3, and #6, especially between #3 and #6. In 9-tone systems, recognition is very difficult since there are so many level tones. 6 of the 9 tones (i.e. #1, #3, #6, #7, #8, #9) in these experiments are level tones. For this reason, it is no surprise to observe low recognition for the level tones.

3.13 Results by Speaker

Table 3-35 shows recognition results for individual speakers for Speaker-Dependent experiments. More detail can be found in the appendices. Analysis of the average recognition rate for speaker-dependent experiments shows which speaker's tones are consistent when compared to other tones from the same speaker. Speaker cs01f achieves the highest score with an average of 90.9%. This suggests that this speaker is most consistent as her tones best matched her other samples. Speaker cs03f, on the other hand, is the most inconsistent, as shown by the recognizer's difficulty in recognizing her tones.

	6-Tone/ 8-Feature	6-Tone/ 16-Feature	9-Tone/ 8-Feature	9-Tone/ 16-Feature	AVG
cs01f	94.5	95.0	79.5	94.5	90.9
cs02m	89.5	90.0	83.0	88.0	87.6
cs03f	89.5	87.5	80.5	79.5	84.3
cs04m	88.5	91.0	80.5	86.0	86.5

Table 3-35 Individual Speaker Recognition Accuracy (Speaker-Dependent)

Individual speaker results for speaker-independent experiments are shown in Table 3-36. Again, the most striking observation is the low results for speaker cs03f, particularly in the 6-tone/8-feature and 9-tone/16-feature systems. In both cases, recognition rate is drastically lower than other speakers. This shows that the tones from cs03f are not very similar to those from the other three speakers. It is not clear why results are not exceptionally poor in the 6-tone/16-feature and 9-tone/8-feature systems. In Appendix C, for instance, tone #4 is recognized very well for the other three speakers (99.2%, 100.0%, and 99.6%). For cs03f, however, accuracy is only 23.5%. Clearly data from cs03f is quite different from data from the others, confirming what was previously suggested in Chapter 2.2.5.

	6-Tone/ 8-Feature	6-Tone/ 16-Feature	9-Tone/ 8-Feature	9-Tone/ 16-Feature	AVG
cs01f	86.6	86.0	76.2	83.9	83.2
cs02m	90.0	85.2	76.7	85.7	84.4
cs03f	62.8	79.7	75.5	58.5	69.1
cs04m	84.1	76.3	65.3	81.9	76.9

Table 3-36 Individual Speaker Recognition Accuracy (Speaker-Independent)

Without data from more speakers, however, it will be unclear as to how erratic data is from this speaker. Manual inspection of several speech samples given by speaker cs03f confirmed that the speaker’s tones are noticeably different in contour than the typical speaker. The developers of CUSYL, however, have made no mention of such idiosyncrasy.

3.14 Mandarin Tone Recognition

After discussing the viability of using memory-based learning for Cantonese tone recognition, it is natural to wonder if this same approach could perform equally well for

Mandarin tone recognition. This section shows that a Mandarin tone recognizer could theoretically perform much better than the Cantonese tone recognizer presented in this thesis.

In the Mandarin tone recognition task there are four tones: high level (1st tone), middle rising (2nd tone), dipping-then-rising tone (3rd tone), and a high falling tone (4th tone). The 1st tone in Mandarin is essentially identical to Cantonese tone #1. The 2nd tone is extremely similar to Cantonese tone #2. Mandarin tone #3 is similar to Cantonese tone #5 in that both tones dip then rise again. The Mandarin 4th tone is similar to Cantonese tone #4 in that both tones fall sharply. The main distinction between the Mandarin 4th tone and Cantonese tone #4 is pitch, not contour.

Simplifying the Cantonese tone recognition task to the four tones noted above produced the results shown below. The figures in Table 3-37 are merely conjecture as the numbers are merely computed by eliminating errors not possible using the hypothetical 4-tone approach.

	Results(%)
Speaker-Dependent, 8-Features	96.3
Speaker-Dependent, 16-Features	98.1
Speaker-Independent, 8-Features	88.9
Speaker-Independent, 16-Features	92.7

Table 3-37 Mandarin Tone Recognition Results (Speculative)

Because Mandarin possesses fewer tones than Cantonese, as well as tones with unique tonal contours, it is expected that the application of memory-based learning to

Mandarin tone recognition would perform extremely well due to increased simplicity of the task.

It has been shown in this chapter that memory-based learning is indeed a viable approach for Cantonese tone recognition. The results that have just been presented in this chapter will now be summarized in the next chapter, along with suggestions for future work using memory-based learning in application to tone recognition.

4 Conclusions

The primary goal of this thesis was to explore memory-based learning as a viable approach for Cantonese tone recognition. 90.9% accuracy was attained for the speaker-dependent 6-tone system, and 87.0% for the 9-tone system. For speaker-independent systems, accuracies of 81.8% and 77.5% were attained for the 6-tone and 9-tone systems respectively. The 90.9% accuracy in speaker-dependent experiments is comparable with the work of Tan Lee et al. (1993,1995) in which 89.0% was reported using a neural network classifier. Thousands of parameter permutations were attempted to find which combination is best for the memory-based learning approach.

Choice of feature extraction method has a considerable effect on recognition accuracy. Auto Correlation was found to be the best method of feature extraction for experiments using only 8 features. This method, however, did not perform as well in experiments using 16 features. Rather, the Cepstrum methods (both CepA and CepB) work best, particularly when using the zero-addition or 100-addition method of feature selection.

A key part of memory-based learning is its classification algorithm. The IB1 algorithm was compared with the popular IGTREE method. As suggested in TiMBL documentation, the IB1 algorithm was shown to produce highest accuracy in this work.

As expected, normalizing fundamental frequencies was found to be essential for speaker-independent experiments. Accuracy increases from 51.4% to 81.0% for 6-tone experiments and 46.8% to 72.5% for 9-tone experiments were observed when the features were normalized. Normalizing data also proved useful in isolating inconsistent speakers (in this study, speaker cs03f).

A common technique in both Cantonese and Mandarin feature-based tone recognition is to trim features on both ends of the feature vector in an attempt to eliminate erratic tone data. When using 8 features, the best results in this thesis were found by not trimming any features off the feature vectors. In fact, a direct correlation was observed in which the more features were ignored, the lower the accuracy. In the speaker-dependent 16-feature system, it was found that ignoring the first two features attained best results in speaker-dependent systems. In speaker-independent 16-feature systems, no significant benefit was observed through either trimming or not trimming.

It appears the choice of feature weighting method makes very little difference in accuracy results. For most recognition systems explored, no significant advantage was observed by any of the feature weighting methods attempted in this study. It was clear, however, that in 9-tone/8-feature systems, not using any feature weighting method resulted in lower accuracy than any of the methods for feature weighting. With only a couple of small exceptions, using the Gain Ratio method achieved highest recognition accuracy albeit by very small margins.

In calculating distances between features, *k*-values determine how many nearest-neighbors to consider in the computation. The general trend observed in this work is that the higher the value of *k*, the better the accuracy. This is especially true in speaker-independent systems. It could be that using a value *k*-value higher than 15 could result in even better results; maximum accuracy of several experiments was found when the value of *k* was set to 15. Further empirical work could be done to determine the optimum value of *k* for these experiments.

Intertwined with *k*-values is the class voting weights method. Each method implemented performed similarly. The Inverse Linear and Inverse Distance methods achieved the best results. Inverse Linear is the best method for 6-tone systems, whereas Inverse Distance is the best for 9-tone systems. Speaker-independent systems tended to favor the Inverse Linear approach.

It is not clear how many features are optimal when extracting fundamental frequencies from a speech item. Clearly, too few features don't provide enough information, whereas too many features provides too much. In this study, 16-feature experiments were performed in order to facilitate comparison with other tone recognition techniques, since several researchers have arbitrarily chosen to use 16 features. Experiments were performed using 8 features to see how recognition rate is affected by using a smaller, and less computationally costly, feature vector.

Average recognition for 16-feature systems was 84.3%; average recognition for 8-feature systems was 81.4%. The difference was much greater for 9-tone systems than for 6-tone systems. Using 8 features instead of 16 resulted in a loss of only 0.4% in the speaker-dependent 6-tone system and 0.9% in the speaker-independent 6-tone system. In 9-tone systems, however, losses of 6.1% and 4.1% are observed. If computational restrictions do not exist, then clearly using 16 features is better. Using 8 features is a viable option in that it will lessen computational burden with only a slight loss of accuracy, especially in 6-tone recognition systems.

A very important decision to make when designing a Cantonese tone recognizer is how many tones to categorize between. Because the number of tones in Cantonese is debatable, several possibilities exist. In this thesis, a comparison between 6-tone systems and 9-tone systems was performed. The average result attained in the 9-tone systems was 79.7%, whereas 6-tone systems achieved 86.0%. The fact that higher accuracy was obtained using a 6-tone system is not surprising as the additional three tones used in a 9-tone system are extremely similar to existing tones in the 6-tone system. In addition, extra tones introduce more opportunities for misclassification. Unless the need exists for a 9-tone system, the use of a 6-tone system is strongly recommended.

Results from speaker-dependent experiments were found to be much higher than those from speaker-independent experiments. The average result attained in speaker-dependent experiments was 87.3%, whereas average results were 78.4% on speaker-independent experiments. The speaker-independent results, however, are not truly speaker-independent in that average tone frequency for each speaker was predetermined and used in the normalization process. A true speaker-independent system could, however, normalize data online during processing or be precomputed with only a minimal amount of speaker training.

For each of the 8 systems developed in this thesis, a different combination of parameters produced the highest results. Table 4-1 and Table 4-2 are reproduced here to summarize which parameters attained best results for each of the 8 systems.

	FE	Trim	FW	*k*	CVW	%
Speaker-Dependent 6-Tone/8-Feature	AC	---	gr	13	ID	90.5
Speaker-Dependent 9-Tone/8-Feature	CepB	---	gr	11/13	IL	80.9
Speaker-Dependent 6-Tone/16-Feature	CepB(0)	I1-2	sv/χ^2	5	ID	90.9
Speaker-Dependent 9-Tone/16-Feature	CepA(100)	I1-2	gr	11	IL	87.0
Speaker-Independent 6-Tone/8-Feature	CepA	---	sv	15	ID	80.9
Speaker-Independent 9-Tone/8-Feature	AC	---	gr	15	IL	73.4
Speaker-Independent 6-Tone/16-Feature	CepA(s)	---	sv/χ^2	13	ID	81.8
Speaker-Independent 9-Tone/16-Feature	CepA(0)	I1-2	gr	15	ID	77.5

Table 4-1 Parameters producing best results for all 8 systems

FE	Feature Extraction Method
Trim.	Features Trimmed
FW	Feature Weighting Method
CVW	Class Voting Weighting Method
AC	Auto Correlation
CepB	CepstrumB
CepB(0)	CepstrumB(Zero-Addition)
CepA(100)	CepstrumA(100-Addition)
CepA	CepstrumA
CepA(s)	CepstrumA(Stretched)
CepA(0)	CepstrumA(Zero-Addition)
I1-2	Ignore 1st and 2nd feature
gr	Gain Ratio
sv	Shared Variance
χ^2	Chi-squared
ID	Inverse Distance
IL	Inverse Linear

Table 4-2 Legend of Abbreviations for Table 4-1

Analysis of confusion matrices helped isolate which tones were particularly easy or difficult to recognize. It also provided information as to which tone pairs are most easily confused. In speaker-dependent experiments, tones #1, #2, and #4 were all recognized much easier than the other tones. Tone #5, on the other hand, was often misclassified. In speaker-independent experiments, tone #1 was recognized extremely well.

In all experiments, the most common classification error was due to the recognizer's difficulty in distinguishing among level tones. In 6-tone systems, tones #1, #3, and #6 were often confused, especially between tone #3 and #6. The addition of three more level tones in 9-tone systems added to the confusion, especially between tones that differed primarily by duration rather than contour or pitch. Overall, tone #1 was the easiest tone to classify. This is surprising as it is a level tone and predictably would be confused with the other 5 level tones (i.e. tones #3, #6, #7, #8, and #9). It is not clear why tone #1 is recognized easier than other tones, especially those with unique contours.

One major limitation throughout this study was the scarcity of data. Only 2000 files from 4 different speakers were available. One speaker in particular, cs03f, speaks very differently than the other three speakers. Results from this speaker were poor in both speaker-dependent and speaker-independent systems. This suggests the speaker is not only inconsistent when compared to other speakers, but is inconsistent even when compared with her own speech. With data from only four speakers available, it is unclear how idiosyncratic this speaker's data is.

It is expected the approach used in this thesis would achieve even higher results when applied to the Mandarin tone recognition task. Not only are there only four tones in Mandarin, but also the contours of each tone are more distinct from each other. Limiting the tone recognition task to the four Cantonese tones most similar to the four Mandarin tones produced hypothetical results for a Mandarin tone recognizer. Hypothetical results are 98.1% for a speaker-dependent system, and 92.7% for a speaker-independent system.

Results from using the memory-based learning approach in this study are comparable to the work of Tan Lee et al. (1993, 1995), in which a neural-network was applied to a speaker-dependent 9-tone system. In Lee's work, recognition accuracy of 89.0% was achieved in a speaker-dependent system. Using the memory-based learning approach as presented in this thesis, recognition rate was 90.9%. Again, one of the four speakers in this study exhibited abnormal tonal contours and her recognition accuracy was extremely poor in comparison to results from other speakers. Data from this speaker might also have weakened the training set used for recognition for other speakers' data.

As a result, the recognition rate of 90.9% could be much higher depending on how abnormal the data is from this one speaker.

The system presented in this thesis can be improved upon in many ways: 1) finding and using an optimal number of features in each feature vector; 2) increasing the amount of training data; 3) using higher *k*-values; 4) using a more accurate feature extraction method; 5) adding duration elements to the process for 9-tone experiments.

This thesis shows that memory-based learning is a viable option for Cantonese tone recognition. It also shows that absolute values can be useful to model tones instead of using only contours. Memory-based approaches are bound to work increasingly better as computers get more powerful.

Memory-based learning, as discussed in this thesis, could be applied to the recognition of other tonal languages such as Mandarin, Thai, Vietnamese, or other dialects of Chinese. Work in the future might also apply this approach to more complicated tasks such as continuous speech.

REFERENCES

Aha, David. W. 1998. Feature weighting for lazy learning algorithms. In Feature Extraction, Construction and Selection: A Data Mining Perspective, ed. H. Liu and H. Motoda. Norwell, MA: Kluwer.

Androutsopoulos, Ion, Georgios Paliouras, Vangelis Karkaletsis, Georgis Sakkis, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos. 2000. Learning to filter spam e-mail: A comparison of a naïve Bayesian and a memory-based approach. In Proceedings of the "Machine Learning and Textual Information Access" Workshop of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (KDD).

Bauer, Robert S. and Paul K. Benedict. 1997. Modern Cantonese phonology. New York: Mouton de Gruyter.

Buchholz, Sabine. 1998a. Unsupervised learning of subcategorisation information and its application in a parsing subtask. In Proceedings of the Tenth Netherlands/Belgium Conference on Artificial Intelligence, ed. by H. La Poutre and H.J. van den Herik, CWI, Amsterdam. 7-16.

Buchholz, Sabine. 1998b. Distinguishing complements from adjuncts using memory-based learning. In Proceedings of the Workshop on Automated Acquisition of Syntax and

Parsing, 10th European Summer School in Logic, Language and Information (ESSLLI). Saarbrücken, Germany.

Buchholz, Sabine, Jorn Veenstra, and Walter Daelemans, 1999. Cascaded grammatical relation assignment. In Proceedings of the 1999 Conference on Empirical Methods in Natural Language Processing (EMNLP). College Park, MD. University of Maryland. 239-246.

Burton, David K., John E. Shore, and Joseph T. Buck. 1985. Isolated-word speech recognition using multisection vector quantization codebooks. In IEEE Transactions on Acoustics, Speech and Signal Processing 33. 837-849.

Busser, Bertjan, Walter Daelemans and Antal van den Bosch. 1999. Machine learning of word pronunciation: the case against abstraction. In Proceedings of the Sixth European Conference on Speech Communication and Technology (Eurospeech99). 2123-2126.

Cardie, Claire. 1996. Automatic feature set selection for case-based learning of linguistic knowledge. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). University of Pennsylvania. 113-126.

Cardie, Claire. 1994. Domain-specific knowledge acquisition for conceptual sentence analysis, Ph.D. Thesis, University of Massachusetts, Amherst, MA.

Chang, Kai-Cheng and C.C. Yang. 1986. A real-time pitch extraction and four-tone recognition system for Mandarin speech. Journal of the Chinese Institute of Engineers. 37-49.

Chang, Pao-Chung, San-Wei Sun, and Sin-Horng Chen. 1990. Mandarin tone recognition by multi-layer perceptron. In Proceedings of the 1990 International Conference on Acoustics, Speech and Signal Processing (ICASSP). 517-520.

Chao, Yuen Ren. 1947. Cantonese Primer. Greenwood Press Publishers.

Chen, Sin-Horng and Yih-Ru Wang. 1995. Tone recognition of continuous Mandarin speech based on neural networks. IEEE Transactions on Speech and Audio Processing 3. 146-150.

Chen, Xi-Xian, Chang-Nian Cai, Peng Guo, and Ying Sun. 1987. A hidden Markov model applied to Chinese four-tone recognition. In Proceedings of the 1987 International Conference on Acoustics, Speech and Signal Processing (ICASSP). 797-800.

Ching, P.C., Tan Lee, and Eric Zee. 1994. From phonology and acoustic properties to automatic recognition of Cantonese. In Proceedings from 1994 International Syposium on Speech, Image Processing, and Neural Networks (ISSIPNN) 1. 127-132.

Chow, K.F., Tan Lee, and P.C. Ching. 1998. Sub-syllable acoustic modeling for Cantonese speech recognition. In Proceedings of 1998 International Symposium on Chinese Spoken Language Processing (ISCSLP). 75-9.

Cost, S. and S. Salzberg. 1993. A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning 10. 57-78.

Cox, S.J. 1990. Hidden Markov models for automatic speech recognition: Theory and application. Speech and Language Processing, ed. by C. Whedden and R. Linggard. 209-30.

Daelemans, Walter. 1998. Toward an exemplar-based computational model for cognitive grammar. English as a Human Language, ed. by Van Der Auwera et al. Munich: LINCOM. 73-82.

Daelemans, Walter, and Antal van den Bosch. 1992. Generalisation performance of backpropagation learning on a syllabification task. In Proceedings of TWLT3: Connectionism and Natural Language Processing, ed. by M.F.J. Drossaers and A. Nijholt. 27-37.

Daelemans, Walter, Steven Gillis, and Gert Durieux. 1994. The acquisition of stress: a data-oriented approach. Computational Linguistics 20. 421-451.

Daelemans, Walter, Peter Berck, and Steven Gillis. 1997. Data mining as a method for linguistic analysis: Dutch diminutives. Folia Linguistica, XXXI(1-2).

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2001. TiMBL: Tilburg Memory Based Learner, version 4.1, Reference Guide. ILK Technical Report 01-04, Available from http://ilk.kub.nl/downloads/pub/papers/ilk0104.ps.gz

Daelemans, Walter, Antal van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. Machine Learning 34. 11-43.

Daelemans, Walter, Antal van den Bosch, Jakub Zavrel, Jorn Veenstra, Sabine Buchholz, and Bertjan Busser. 1998a. Rapid development of NLP modules with memory-based learning. In Proceedings of ELSNET in Wonderland. Utrecht: ELSNET. 105-113. Also in ECML-98 TANLPS Workshop Notes, ed. by r. Basili and M.T. Pazienza. 1998. Technische Universitaet Chemnitz. 1-17.

Daelemans, Walter, Gert Durieux, and Antal van den Bosch. 1998b. Toward inductive lexicons: a case study. In Proceedings of the Workshop on Adapting Lexical and Corpus Resources to Sublanguages and Applications, International Conference on Language Resources and Evaluation (LREC), ed. by P. Velardi. 29-35.

Daelemans, Walter, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: A memory-based part of speech tagger-generator. In Fourth Workshop on Very Large Corpora, ed. by E. Ejerhed and I. Dagan. Copenhagen. 14-27.

de Cheveigné, Alain, Hideki Kawahara. 2001. Comparative evaluation of F_0 estimation algorithms. In Proceedings of the Seventh European Conference on Speech Communication and Technology (Eurospeech 2001). 2451-2454.

Devijver, P.A., J. Kittler. 1980. On the edited nearest neighbor rule. In Proceedings of the Fifth International Conference on Pattern Recognition. 72-80.

Dudani, S.A. 1976. The distance-weighted *k*-nearest neighbor rule. In IEEE Transactions on Systems, Man, and Cybernetics SMC-6. 325-327.

Durieux, Gert, and Steven Gillis. 2000. Predicting grammatical classes from phonological cues: An empirical test. In Approaches to Bootstrapping: Phonological Syntactic and Neurophysiological Aspects of Early Language Acquisition, ed. by B. Höhle and J. Weissenborn. Benjamins, Amsterdam. 189-232.

Gao, Sheng, Tan Lee, Y.W. Wong, Bo Xu, P.C. Ching, and Taiyi Huang. 2000. Acoustic modeling for Chinese speech recognition: A comparative study of Mandarin and Cantonese. In Proceedings of the 2000 International Conference on Acoustics, Speech and Signal Processing (ICASSP). Istanbul, Turkey. 1261-4.

Gates, G.W. 1972. The reduced nearest neighbor rule. IEEE Transactions on Information Theory 18. 431-433.

Gillis, Steven, Gert Durieux, and Walter Daelemans. 2000. Lazy learning: A comparison of natural and machine learning of stress. In Models of Language Acquisition: inductive and deductive approaches, ed. by P. Broeder and J.M.J. Murre. Oxford University Press. 76-99.

Gray, Robert M. 1990. Vector quantization. Readings in Speech Recognition, ed. by Alex Waibel and Kai-Fu Lee. San Mateo, CA: Morgan Kaufmann. 75-100.

Guan, Cuntai and Chen Yongbin. 1993. Speaker-independent tone recognition for Chinese speech. Acta Acustica 18. 380-385.

Gustafson, Joakim, Nikolas Lindberg, and Magnus Lundeberg. 1999. The August spoken dialogue system. In Proceedings of the European Conference on Speech Technology. (Eurospeech'99).

Hart, P.E. 1968. The condensed nearest neighbor rule. IEEE Transactions on Information Theory 16. 515-516.

Hernández-Ábrego, Gustavo A. 2000. Confidence measures for speech recognition and utterance verification. PhD. Thesis. Departament de Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya.

Huang, Xuedong and Kai-Fu Lee. 1993. On speaker-independent, speaker-dependent, and speaker-adaptive speech recognition. IEEE Transactions on Speech and Audio Processing 1. 150-157.

Huang, Xuedong, Alex Acero, and Hsiao-Wuen Hon. 2001. Spoken language processing: A guide to theory, algorithm, and system development. New Jersey: Prentice Hall.

Huckvale, Mark. 2000. Speech Filing System Documentation (Help Contents). http://www.phon.ucl.ac.uk/resource/sfs/help/index.htm

Jelinek, Frederick. 1997. Statistical methods for speech recognition. Cambridge: MIT Press.

Kocsor, Andrá, László Tóth, András Kuba Jr., Kornél Kovács, Márk Jelaity, Tibor Gyimóthy, and Jánds Csirik. 2000. A comparative study of several feature transformation and learning methods for phoneme classification. International Journal of Speech Technology 3. 263-276.

Krahmer, E., Marc Swerts, M. Theune, and M. Weegels. 2001. Error detection in spoken human-machine interaction. International Journal of Speech Technology 4. 19-30.

Krott, Andrea, R. Harald Baayen, and Robert Schreuder. 2001. Analogy in morphology: modeling the choice of linking morphemes in Dutch. Linguistics 39. 51-93.

Lau, Wai, Y.W. Wong, W.K. Lo, Tan Lee, and P.C. Ching. 2000a. A study on the contribution of lexical tones in Chinese LVCSR. In Proceedings of the 2000 International Symposium on Chinese Spoken Language Processing (ISCSLP). 129-132.

Lau, Wai, Tan Lee, Y.W. Wong, and P.C. Ching. 2000b. Incorporating tone information into Cantonese large-vocabulary continuous speech recognition. In Proceedings of the 2000 International Conference on Spoken Language Processing (ICSLP) 2.883-886.

Lee, Tan and P.C. Ching. 1999. Cantonese syllable recognition using neural networks. IEEE Transactions on Speech and Audio Processing 7. 466-72.

Lee, Tan and P.C. Ching. 1997. A neural network based speech recognition system for isolated Cantonese syllables. In Proceedings of 1997 International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Lee, Tan, Greg Kochanski, Chilin Shih, and Yujia Li. 2002a. Modeling tones in continuous Cantonese speech. In Proceedings of the 2002 International Conference on Spoken Language Processing (ICSLP). Denver. 2401-2404.

Lee, Tan, W.K. Lo, and P.C. Ching. 2002b. Spoken language resources for Cantonese speech processing. In Speech Communication 36. 327-342.

Lee, Tan, P.C. Ching, L.W. Chan, Brian Mak, and Y.H.Cheng. 1995. Tone recognition of isolated Cantonese syllables. IEEE Transactions on Speech and Audio Processing 3. 204-9.

Lee, Tan, P.C. Ching, L.W. Chan, and Brian Mak. 1993. An NN based tone classifier for Cantonese. In Proceedings of 1993 International Joint Conference on Neural Networks (IJCNN) 1. Nagoya. 287 –290.

Lin, Chih-Heng, Lin-Shan Lee, and Pei-Yih Ting. 1993. A new framework for recognition of Mandarin syllables with tones using sub-syllabic units. In Proceedings of the 1993 International Conference on Acoustics, Speech and Signal Processing (ICASSP). 227-229.

Lippman, Richard P. 1990. Review of Neural Networks for Speech Recognition. Readings in Speech Recognition, ed. by Alex Waibel and Kai-Fu Lee. San Mateo, CA: Morgan Kaufmann. 374-92.

Liu, Lih-Cherng and Hsiao-Chuan Wang. 1988. Recognition of Mandarin consonants using statistical models. In Proceedings of 1988 International Conference on Computer Processing of Chinese and Oriental Languages (ICCPCOL). 566-570.

Liu, W. Yang, H. Wang, and Y. Chang. 1989. Tone recognition of polysyllabic words in Mandarin speech. Computer Speech and Language 3. 253-264.

Lo, W.K., Tan Lee, and P.C. Ching. 1998. Development of Cantonese spoken language corpora for speech applications. In Proceedings of 1998 International Symposium on Chinese Spoken Language Processing (ISCSLP). 102-7.

Ma, Hengjie. 1987. The four tones recognition of continuous chinese speech. In Proceedings of the 1987 International Conference on Acoustics, Speech and Signal Processing (ICASSP). 65-68.

Matthews, Stephen and Virginia Yip. 1994. Cantonese: A comprehensive grammar. London: Rutledge.

Mendel, Jerry M. 1995. Fuzzy logic systems for engineering: A tutorial. In Proceedings of the IEEE 83. March. 345-77.

Minnen, Guido, Francis Bond, and Ann Copestake. 2000. Memory-based learning for article generation. In Proceedings of the 4th Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop. 43-8.

Ng, Alfred Y.P., Tan Lee, P.C. Ching, and L.W. Chan. 1996. Recent advances in Cantonese speech recognition. In Proceedings of 1996 International Symposium on Multi-Technology Information Processing. December. Hsinchu, Taiwan. 139-44.

Noll, A.M. 1967. Cepstral pitch determination. Journal of the Acoustical Society of America 41. 293-309.

Norman, Jerry. 1988. Chinese. Cambridge: Cambridge University Press.

Orasan, Constantin. 2000. A hybrid method for clause splitting in unrestricted English texts. In Proceedings of the 2000 International Conference on Artificial and Computational Intelligence for Control, Automation and Decision in Engineering and Industrial Systems (ACIDCA), Monastir, Tunisia.

O'Shaughnessy, Douglas. 1987. Speech Communications: Human and Machine. New York: Addison-Wesley Publishing Company.

Ostendorf, Mari. 1996. From HMMs to segment models: Stochastic modeling for CSR. Automatic Speech and Speaker Recognition: Advanced Topics, ed. by Chin-Hui Lee, Frank K. Soong, and Kuldip K. Paliwal. 185-210. Boston, MA: Kluwer.

Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 1992. Numerical recipes in C: the art of scientific computing (2nd ed). Cambridge: Cambridge University Press.

Quinlan, J.R. 1993. C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufman.

Rabiner, Lawrence R. 1990. A tutorial on hidden Markov models and selected applications in speech recognition. Readings in Speech Recognition, ed. by Alex Waibel and Kai-Fu Lee. San Mateo, CA: Morgan Kaufmann. 267-96.

Rabiner, Lawrence R., B.H. Juang, and C.H. Lee. 1996. An overview of automatic speech recognition. Automatic Speech and Speaker Recognition: Advanced Topics, ed. by Chin-Hui Lee, Frank K. Soong, and Kuldip K. Paliwal. 1-30. Boston, MA: Kluwer.

Rose, Richard C. 1996. Word spotting from continuous speech utterances. Automatic Speech and Speaker Recognition: Advanced Topics, ed. by Chin-Hui Lee, Frank K. Soong, and Kuldip K. Paliwal. 303-29. Boston, MA: Kluwer.

Shannon, Claude E. 1960. Coding theorems for a discrete source with a fidelity criterion. Information and Decision Processes, ed. by R.E. Machol. 93-126. New York: New York.

Skousen, Royal. 1992. Analogy and structure. Dordrecht: Kluwer.

Skousen, Royal. 1989. Analogical modeling of language. Dordrecht: Kluwer.

Skousen, Royal, Deryle Lonsdale, and Dilworth Parkinson (ed.). 2002. Analogical modeling: An exemplar-based approach to language. John Benjamins:Amsterdam.

Stanfill, C. and D. Waltz. 1986. Toward memory-based reasoning. Communications of the ACM 29. 1213-28.

Stevenson, Mark, and Robert Gaizauskas. 2000. Experiments on sentence boundary detection. In Proceedings of the Sixth Conference on Applied Natural Language Processing and the First Conference of the North American Chapter of the Association for Computational Linguistics (IJCAI). 24-30.

Tjong Kim Sang, Erik. 2001. Memory-based clause identification. In Proceedings of CoNLL-2001. Toulouse, France. 67-69.

Tjong Kim Sang, Erik and Jorn Veenstra. 1999. Representing text chunks. In Proceedings of the 1999 European Chapter of the Association for Computational Linguistics. Bergen, Norway. 173-9.

Tungthangthum, A. 1998. Tone recognition for Thai. In Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and System. 157-60.

van den Bosch, Antal and Zavrel, Jakub. 2000. Unpacking multi-valued symbolic features and classes in memory-based language learning. In Proceedings of the Seventeenth International Conference on Machine Learning, ed. by P. Langley. San Francisco, CA: Morgan Kaufmann.1055-62.

van den Bosch, Antal. 1999a. Careful abstraction from instance families in memory-based language learning. Journal of Experimental and Theoretical Artificial Intelligence, Special Issue on Memory-Based Language Processing, ed. by W. Daelemans. 11. 339-68.

van den Bosch, Antal. 1999b. Instance-family abstraction in memory-based language learning. In Machine Learning: Proceedings of the Sixteenth International Conference, ICML'99, by I. Bratko and S. Dzeroski. 39-48.

van den Bosch, Antal, and Walter Daelemans. 1993. Data-oriented methods for grapheme-to-phoneme conversion. In Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics. 45-53.

van den Bosch, Antal, and Walter Daelemans. 1999. Memory-based morphological analysis. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL). 285-92.

van den Bosch, Antal, and Walter Daelemans. 2000. A distributed, yet symbolic model of text-to-speech processing. In Models of Language Acquisition: inductive and deductive approaches, ed. by P. Broeder and J.M.J. Murre. Oxford University Press. 55-75.

van den Bosch, Antal, E. Krahmer, and Marc Swerts. 2001. Detecting problematic turns in human-machine interactions: Rule-induction versus memory-based learning approaches. In Proceedings of the 39th Meeting of the Association for Computational Linguistics (ACL). 499-506.

van Helteren, Hans, Jakub Havrel, and Walter Daelemans. 2001. Improving accuracy in word class tagging through combination of machine learning systems. Computational Linguistics. 27. 199-230.

Veenstra, Jorn. 1998. Fast NP chunking using memory-based learning techniques. In Proceedings of Benelearn'98.

Veenstra, Jorn, Antal van den Bosch, Sabine Buchholz, Walter Daelemans, and Jakub Zavrel. 2000. Memory-based word sense disambiguation. Computers and the Humanities. 34(1-2).

Veenstra, Jorn, Antal van den Bosch, Sabine Buchholz, Walter Daelemans, Jakub Zavrel. 1998. Memory-based word sense disambiguation. Computing and the Humanities, Special issue on SENSEVAL.

Wagner, Michael, Wei Wang, and Helen Ho. 1986. Isolated word recognition of the complete vocabulary of spoken Chinese. In Proceedings of the 1986 International Conference on Acoustics, Speech and Signal Processing (ICASSP). 701-4.

Wang, Jhing-Fa, Chung-Hsien Wu, Shih-Hung Chang, and Jau-Yien Lee. 1991. A hierarchical neural network model based on a C/V segmentation algorithm for isolated Mandarin speech recognition. IEEE Transactions on Signal Processing 39. 2141-6.

Wang, Jhing-Fa, Chung-Hsien Wu, Jau-Yien Lee, and Chia-Nien Wang. 1988. Mandarin syllable recognition system with learning ability based on neural network model. In Proceedings of 1988 International Conference on Computer Processing of Chinese and Oriental Languages (ICCPCOL). 513-7.

Wang, Xia and Juha Iso-Sipilä. 2002. Low complexity Mandarin speaker-independent isolated word recognition. In Proceedings of the 2002 International Conference on Spoken Language Processing (ICSLP). Taipei. 1589-92.

Wang, Yih-Ru, Jyh-Ming Shieh, and Sin-Horng Chen. 1994. Tone recognition of continuous Mandarin speech based on hidden Markov model. International Journal of Pattern Recognition and Artificial Intelligence 8. 233-45.

Wettschereck, D. 1994. A study of distance-based machine learning algorithms. Ph.d thesis, Oregon State University.

Wettschereck, D., D.W. Aha, and T. Mohri. 1996. A review and comparative evalation of feature weighting methods for lazy learning algorithms. Technical Report AIC-95-012. Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.

White, Allan P., Wei Zhong Liu. 1994. Bias in information-based measures in decision tree induction. Machine Learning 15. 321-9.

Wilson, D. 1972. Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics 2. 408-21.

Wong, Y.W., K.F. Chow, W. Lau, W.K. Lo, Tan Lee and P.C. Ching. 1999. Acoustic modeling and language modeling for Cantonese LVCSR. In Proceedings of the 6th European Conference on Speech Communication and Technology 3. 1091-4.

Wu, Yadong, Kazou Hemmi, and Kazuo Inoue. 1991. A tone recognition of polysyllabic Chinese words using an approximation model of four tone pitch patterns. In Proceedings of the 1991 International Conference on Industrial Electronics, Control and Instrumentation. 2115-7.

Xu, Shilin and Samuel C. Lee. 1992. A fast real time Chinese tone recognition system using Fuzzy Sets. International Journal of the Chinese & Oriental Languages Information Processing Society 2. 1-13.

Yang, Wu-Ji , Jyh-Chyang Lee, Yuen-Chin Chang, and Hsiao-Chuan Wang. 1988. Recognition of lexical tones for isolated syllables and disyllables in Mandarin speech. International Journal of Pattern Recognition and Artificial Intelligence 2. 49-69.

Yang, Wu-Ji, Jyh-Chyang Lee, Yueh-Chin Chang, and Hsiao-Chuan Wang. 1988a. Hidden Markov model for Mandarin lexical tone recognition. IEEE Transaction on ASSP. 988-92.

Zadeh, Lotfi. 1965. Fuzzy sets. Information and Control 8. 338-53.

Zavrel, Jakub, Peter Berck, and Willem Lavrijssen. 2000. Information extraction by text classification: Corpus mining for features. In Proceedings of the Workshop on Information Extraction meets Corpus Linguistics, The Second International Conference of Language Resources and Evaluation (LREC).

Zavrel, Jakub, and Walter Daelemans. 1999. Recent advances in memory-based part-of-speech tagging. In VI Simposio Internacional de Comunicacion Social. 590-7.

Zavrel, Jakub, Walter Daelemans, and Jorn Veenstra. 1997. Resolving PP attachment ambiguities with memory-based learning. In Proceedings of the Workshop on Computational Natural Language Learning (CoNLL), ed. by M. Ellison.

Zhang, Jin-song and Keikichi Hirose. 2000. Anchoring hypothesis and its application to tone recognition of Chinese continuous speech. In Proceedings of the 2000 International Conference on Acoustics, Speech and Signal Processing (ICASSP). Istanbul. 1419-22.

Zhang, Jin-song and Keikichi Hirose. 1998. A robust tone recognition method of Chinese based on sub-syllabic F0 contours. In Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP) 3. 703-6.

Zhang, Yaxin, Anton Medievski, James Lawrence, Jianming Song. 2000. A study on tone statistics in Chinese names. Speech Communication 36. 267-75.

Zhou, Liang and Satoshi Imai. 1996. Chinese all syllable recognition using combination of multiple classifiers. In Proceedings of the 1996 International Conference on Acoustics, Speech and Signal Processing (ICASSP). 3494-7.

APPENDIX A

	1	2	3	4	5	6	%
1	40	0	0	0	0	1	97.6%
2	0	37	0	0	2	0	94.9%
3	0	0	40	0	0	3	93.0%
4	0	0	0	26	0	0	100.0%
5	0	0	1	0	12	2	80.0%
6	0	0	2	0	0	34	94.4%
						AVG	94.5%

Table 4-3 Speaker cs01f Confusion Matrix (Speaker-Dependent; 6-tone/8-feature)

	1	2	3	4	5	6	%
1	38	0	1	1	0	1	92.7%
2	0	36	0	0	3	0	92.3%
3	1	0	36	0	0	6	83.7%
4	2	0	0	24	0	0	92.3%
5	0	1	0	0	14	0	93.3%
6	0	0	5	0	0	31	86.1%
						AVG	89.5%

Table 4-4 Speaker cs02m Confusion Matrix (Speaker-Dependent; 6-tone/8-feature)

	1	2	3	4	5	6	%
1	40	0	0	0	0	1	97.6%
2	0	37	0	0	2	0	94.9%
3	0	0	39	0	0	4	90.7%
4	1	0	0	24	1	0	92.3%
5	0	0	2	3	9	1	60.0%
6	0	0	6	0	0	30	83.3%
						AVG	89.5%

Table 4-5 Speaker cs03f Confusion Matrix (Speaker-Dependent; 6-tone/8-feature)

	1	2	3	4	5	6	%
1	37	0	4	0	0	0	90.2%
2	0	37	1	0	1	0	94.9%
3	2	0	35	0	0	6	81.4%
4	0	0	0	25	0	1	96.2%
5	0	2	0	0	13	0	86.7%
6	0	0	3	3	0	30	83.3%
							88.5%

Table 4-6 Speaker cs04m Confusion Matrix (Speaker-Dependent; 6-tone/8-feature)

APPENDIX B

	1	2	3	4	5	6	%
1	41	0	0	0	0	0	100.0%
2	0	36	0	0	2	1	92.3%
3	0	0	38	0	1	3	90.5%
4	0	0	0	26	0	0	100.0%
5	0	0	0	0	15	0	100.0%
6	0	0	3	0	0	34	91.9%
						AVG	95.0%

Table 4-7 Speaker cs01f Confusion Matrix (Speaker-Dependent, 6-tone/16-feature)

	1	2	3	4	5	6	%
1	40	0	1	0	0	0	97.6%
2	0	36	0	0	3	0	92.3%
3	2	0	33	0	0	7	78.6%
4	0	0	0	26	0	0	100.0%
5	0	2	0	0	12	1	80.0%
6	0	0	4	0	0	33	89.2%
						AVG	90.0%

Table 4-8 Speaker cs02m Confusion Matrix (Speaker-Dependent, 6-tone/16-feature)

	1	2	3	4	5	6	%
1	41	0	0	0	0	0	100.0%
2	0	36	0	1	1	1	92.3%
3	0	0	38	0	0	4	90.5%
4	0	0	0	25	0	1	96.2%
5	0	3	1	1	7	3	46.7%
6	0	0	8	0	1	28	75.7%
						AVG	87.5%

Table 4-9 Speaker cs03f Confusion Matrix (Speaker-Dependent, 6-tone/16-feature)

	1	2	3	4	5	6	%
1	39	0	2	0	0	0	95.1%
2	0	38	1	0	0	0	97.4%
3	1	0	34	0	0	7	81.0%
4	0	0	0	26	0	0	100.0%
5	0	3	0	0	12	0	80.0%
6	0	0	2	2	0	33	89.2%
						AVG	91.0%

Table 4-10 Speaker cs04m Confusion Matrix (Speaker-Dependent, 6-tone/16-feature)

APPENDIX C

	1	2	3	4	5	6	%
1	414	7	1	0	0	0	98.1%
2	0	284	0	6	22	4	89.9%
3	9	1	233	0	0	106	66.8%
4	0	0	0	245	0	2	99.2%
5	0	9	0	6	107	14	78.7%
6	0	0	7	46	0	269	83.5%
						AVG	86.6%

Table 4-11 Speaker cs01f Confusion Matrix (Speaker-Independent, 6-tone/8-feature)

	1	2	3	4	5	6	%
1	402	0	20	0	0	0	95.3%
2	1	302	0	1	10	2	95.6%
3	15	0	301	0	1	32	86.2%
4	0	0	0	247	0	0	100.0%
5	0	19	0	4	113	0	83.1%
6	0	1	44	17	12	248	77.0%
						AVG	90.0%

Table 4-12 Speaker cs02m Confusion Matrix (Speaker-Independent, 6-tone/8-feature)

	1	2	3	4	5	6	%
1	411	3	4	3	0	1	97.4%
2	0	118	13	2	102	81	37.3%
3	21	0	303	0	0	25	86.8%
4	0	0	0	58	2	187	23.5%
5	0	0	11	0	27	98	19.9%
6	0	0	114	0	0	208	64.6%
						AVG	62.8%

Table 4-13 Speaker cs03f Confusion Matrix (Speaker-Independent, 6-tone/8-feature)

	1	2	3	4	5	6	%
1	395	0	27	0	0	0	93.6%
2	2	296	4	0	10	4	93.7%
3	16	2	242	1	0	88	69.3%
4	0	0	0	246	0	1	99.6%
5	0	50	0	1	85	0	62.5%
6	0	5	42	28	4	243	75.5%
						AVG	84.1%

Table 4-14 Speaker cs04m Confusion Matrix (Speaker-Independent, 6-tone/8-feature)

APPENDIX D

	1	2	3	4	5	6	%
1	409	0	1	7	0	5	96.9%
2	0	311	2	1	2	0	98.4%
3	54	0	281	0	0	14	80.5%
4	5	0	0	235	0	7	95.1%
5	1	11	13	0	95	16	69.9%
6	2	0	102	4	3	211	65.5%
						AVG	86.0%

Table 4-15 Speaker cs01f Confusion Matrix (Speaker-Independent, 6-tone/16-feature)

	1	2	3	4	5	6	%
1	406	0	11	4	0	1	96.2%
2	0	312	0	0	4	0	98.7%
3	33	0	266	0	0	49	76.4%
4	97	0	0	150	0	0	60.7%
5	1	12	1	1	120	1	88.2%
6	0	0	41	9	0	272	84.5%
						AVG	85.2%

Table 4-16 Speaker cs02m Confusion Matrix (Speaker-Independent, 6-tone/16-feature)

	1	2	3	4	5	6	%
1	416	0	1	2	0	3	98.6%
2	2	289	3	0	19	3	91.5%
3	46	0	268	1	0	34	76.8%
4	13	0	0	119	22	93	48.2%
5	0	2	12	0	77	45	56.6%
6	1	1	56	2	2	260	80.7%
						AVG	79.7%

Table 4-17 Speaker cs03f Confusion Matrix (Speaker-Independent, 6-tone/16-feature)

	1	2	3	4	5	6	%
1	278	1	139	2	0	2	65.9%
2	0	260	0	1	55	0	82.3%
3	4	1	170	0	4	170	48.7%
4	1	0	0	241	0	5	97.6%
5	0	3	0	4	129	0	94.9%
6	0	2	13	12	5	290	90.1%
						AVG	76.3%

Table 4-18 Speaker cs04m Confusion Matrix (Speaker-Independent, 6-tone/16-feature)

APPENDIX E

	1	2	3	4	5	6	7	8	9	%
1	29	0	0	0	0	0	0	0	0	100.0%
2	0	36	0	0	2	0	0	0	1	92.3%
3	0	0	29	0	0	1	0	2	0	90.6%
4	0	0	0	25	0	1	0	0	0	96.2%
5	0	1	0	0	13	1	0	0	0	86.7%
6	0	0	1	0	0	15	0	0	0	93.8%
7	7	0	0	0	0	0	5	0	0	41.7%
8	0	0	8	0	1	0	0	1	1	9.1%
9	0	0	1	0	1	11	0	1	6	30.0%
									AVG	79.5%

Table 4-19 Speaker cs01f Confusion Matrix (Speaker-Dependent, 9-tone/8-feature)

	1	2	3	4	5	6	7	8	9	%
1	28	0	1	0	0	0	0	0	0	96.6%
2	0	39	0	0	0	0	0	0	0	100.0%
3	1	0	25	0	0	5	0	1	0	78.1%
4	0	0	0	26	0	0	0	0	0	100.0%
5	0	2	0	0	12	1	0	0	0	80.0%
6	0	0	2	0	0	11	0	0	3	68.8%
7	3	0	0	0	0	0	9	0	0	75.0%
8	0	1	1	1	0	1	0	5	2	45.5%
9	0	0	2	0	0	6	0	1	11	55.0%
									AVG	83.0%

Table 4-20 Speaker cs02m Confusion Matrix (Speaker-Dependent, 9-tone/8-feature)

	1	2	3	4	5	6	7	8	9	%
1	27	0	0	0	0	0	2	0	0	93.1%
2	0	35	0	1	2	1	0	0	0	89.7%
3	1	0	29	0	0	1	0	0	1	90.6%
4	0	0	0	25	0	1	0	0	0	96.2%
5	0	3	1	1	8	2	0	0	0	53.3%
6	0	0	3	0	2	10	0	0	1	62.5%
7	0	0	0	0	0	0	12	0	0	100.0%
8	0	0	3	0	0	1	0	4	3	36.4%
9	0	0	1	1	0	5	0	2	11	55.0%
									AVG	80.5%

Table 4-21 Speaker cs03f Confusion Matrix (Speaker-Dependent, 9-tone/8-feature)

	1	2	3	4	5	6	7	8	9	%
1	27	0	2	0	0	0	0	0	0	93.1%
2	0	35	0	0	3	0	0	0	1	89.7%
3	1	0	24	0	0	5	0	1	1	75.0%
4	0	0	0	24	0	2	0	0	0	92.3%
5	0	2	0	0	13	0	0	0	0	86.7%
6	0	0	3	1	0	12	0	0	0	75.0%
7	5	0	0	0	0	0	7	0	0	58.3%
8	0	0	4	0	1	0	0	6	0	54.5%
9	0	0	0	1	0	4	0	2	13	65.0%
									AVG	80.5%

Table 4-22 Speaker cs04m Confusion Matrix (Speaker-Dependent, 9-tone/8-feature)

APPENDIX F

	1	2	3	4	5	6	7	8	9	%
1	29	0	0	0	0	0	0	0	0	100.0%
2	0	37	0	0	1	0	0	0	1	94.9%
3	0	0	30	0	0	1	0	0	0	96.8%
4	0	0	0	26	0	0	0	0	0	100.0%
5	0	0	0	0	15	0	0	0	0	100.0%
6	0	0	1	0	0	15	0	0	0	93.8%
7	0	0	0	0	0	0	12	0	0	100.0%
8	0	0	0	0	0	0	0	8	3	72.7%
9	0	0	1	1	0	1	0	1	17	81.0%
									AVG	94.5%

Table 4-23 Speaker cs01f Confusion Matrix (Speaker-Dependent, 9-tone/16-feature)

	1	2	3	4	5	6	7	8	9	%
1	28	0	1	0	0	0	0	0	0	96.6%
2	0	36	0	0	2	0	0	0	1	92.3%
3	1	0	25	0	0	5	0	0	0	80.6%
4	0	0	0	26	0	0	0	0	0	100.0%
5	0	2	0	0	12	1	0	0	0	80.0%
6	0	0	1	0	0	15	0	0	0	93.8%
7	1	0	1	0	0	0	10	0	0	83.3%
8	0	0	0	0	0	1	0	8	2	72.7%
9	0	0	0	0	0	4	0	1	16	76.2%
									AVG	88.0%

Table 4-24 Speaker cs02m Confusion Matrix (Speaker-Dependent, 9-tone/16-feature)

	1	2	3	4	5	6	7	8	9	%
1	26	0	0	0	0	0	3	0	0	89.7%
2	0	27	0	1	7	1	0	0	3	69.2%
3	0	0	30	0	0	1	0	0	0	96.8%
4	0	0	0	25	0	1	0	0	0	96.2%
5	0	4	1	1	7	2	0	0	0	46.7%
6	0	0	2	0	0	13	0	0	1	81.3%
7	1	0	0	0	0	0	11	0	0	91.7%
8	0	0	2	0	0	0	0	7	2	63.6%
9	0	0	0	3	0	3	0	2	13	61.9%
									AVG	79.5%

Table 4-25 Speaker cs03f Confusion Matrix (Speaker-Dependent, 9-tone/16-feature)

	1	2	3	4	5	6	7	8	9	%
1	26	0	2	0	0	0	1	0	0	89.7%
2	0	38	0	0	0	0	0	0	1	97.4%
3	0	0	26	0	0	5	0	0	0	83.9%
4	0	0	0	26	0	0	0	0	0	100.0%
5	0	4	0	0	11	0	0	0	0	73.3%
6	0	0	3	1	0	12	0	0	0	75.0%
7	3	0	0	0	0	0	9	0	0	75.0%
8	0	0	2	0	0	0	0	8	1	72.7%
9	0	0	0	1	0	3	0	1	16	76.2%
									AVG	86.0%

Table 4-26 Speaker cs04m Confusion Matrix (Speaker-Dependent, 9-tone/16-feature)

APPENDIX G

	1	2	3	4	5	6	7	8	9	%
1	293	0	3	5	0	5	20	0	1	89.6%
2	0	313	0	1	1	0	0	1	0	99.1%
3	45	0	158	0	0	15	1	34	0	62.5%
4	5	0	0	230	0	2	0	0	10	93.1%
5	0	18	10	0	88	14	0	1	5	64.7%
6	1	0	57	0	5	122	0	17	14	56.5%
7	28	0	0	2	0	1	64	0	0	67.4%
8	12	0	12	0	0	1	10	58	3	60.4%
9	0	0	17	3	0	11	0	35	40	37.7%
									AVG	76.2%

Table 4-27 Speaker cs01f Confusion Matrix (Speaker-Independent, 9-tone/8-feature)

	1	2	3	4	5	6	7	8	9	%
1	314	0	6	2	0	1	4	0	0	96.0%
2	0	311	0	0	5	0	0	0	0	98.4%
3	37	0	187	0	0	27	0	2	0	73.9%
4	40	0	0	181	0	0	26	0	0	73.3%
5	1	10	1	1	122	1	0	0	0	89.7%
6	0	0	34	4	3	171	0	0	4	79.2%
7	59	0	0	1	0	0	33	2	0	34.7%
8	3	0	52	0	0	13	1	20	6	21.1%
9	0	0	17	6	0	47	0	2	34	32.1%
									AVG	76.7%

Table 4-28 Speaker cs02m Confusion Matrix (Speaker-Independent, 9-tone/8-feature)

	1	2	3	4	5	6	7	8	9	%
1	320	0	2	0	0	3	2	0	0	97.9%
2	1	282	1	0	29	3	0	0	0	89.2%
3	32	0	198	0	0	9	3	10	1	78.3%
4	13	1	0	122	27	51	0	0	33	49.4%
5	0	3	5	0	85	36	0	3	4	62.5%
6	0	0	26	0	7	169	0	6	8	78.2%
7	28	0	0	2	0	0	61	4	0	64.2%
8	3	0	18	0	0	6	2	46	21	47.9%
9	1	1	8	1	0	16	0	9	70	66.0%
									AVG	75.5%

Table 4-29 Speaker cs03f Confusion Matrix (Speaker-Independent, 9-tone/8-feature)

	1	2	3	4	5	6	7	8	9	%
1	185	0	128	2	0	2	9	1	0	56.6%
2	0	254	0	1	61	0	0	0	0	80.4%
3	2	0	94	0	0	125	0	9	23	37.2%
4	1	0	0	235	0	6	0	0	5	95.1%
5	0	5	0	5	126	0	0	0	0	92.6%
6	0	0	9	7	0	192	0	0	8	88.9%
7	34	1	20	2	0	0	26	12	0	27.4%
8	2	1	37	0	7	21	0	12	16	12.5%
9	0	0	5	10	5	35	0	5	46	43.4%
									AVG	65.3%

Table 4-30 Speaker cs04m Confusion Matrix (Speaker-Independent, 9-tone/8-feature)

APPENDIX H

	1	2	3	4	5	6	7	8	9	%
1	320	0	1	0	0	0	6	0	0	97.9%
2	0	280	0	3	29	0	0	0	4	88.6%
3	1	0	174	0	0	76	0	0	1	69.0%
4	0	0	0	246	0	0	0	0	1	99.6%
5	0	8	0	1	115	12	0	0	0	84.6%
6	0	0	3	24	0	189	0	0	0	87.5%
7	7	6	2	0	0	0	80	0	0	84.2%
8	4	0	17	0	0	0	4	29	42	30.2%
9	0	0	3	22	0	11	0	0	70	66.0%
									AVG	83.9%

Table 4-31 Speaker cs01f Confusion Matrix (Speaker-Independent, 9-tone/16-feature)

	1	2	3	4	5	6	7	8	9	%
1	313	0	14	0	0	0	0	0	0	95.7%
2	0	301	0	1	9	0	3	0	2	95.3%
3	13	0	213	0	0	24	0	0	2	84.5%
4	0	0	0	247	0	0	0	0	0	100.0%
5	0	8	0	2	126	0	0	0	0	92.6%
6	0	0	42	4	9	159	0	0	2	73.6%
7	18	0	1	0	0	0	71	5	0	74.7%
8	4	0	37	0	0	4	0	48	3	50.0%
9	0	1	8	5	1	31	0	3	57	53.8%
									AVG	85.7%

Table 4-32 Speaker cs02m Confusion Matrix (Speaker-Independent, 9-tone/16-feature)

	1	2	3	4	5	6	7	8	9	%
1	295	3	1	3	0	0	23	2	0	90.2%
2	0	78	22	2	127	66	4	2	15	24.7%
3	20	0	221	0	0	2	0	8	1	87.7%
4	0	0	0	92	1	98	0	0	56	37.2%
5	0	0	11	0	38	85	0	1	1	27.9%
6	0	0	104	0	0	86	0	4	22	39.8%
7	4	0	0	0	0	0	84	6	1	88.4%
8	0	0	11	1	0	0	2	65	17	67.7%
9	0	0	2	1	0	4	0	10	89	84.0%
									AVG	58.5%

Table 4-33 Speaker cs03f Confusion Matrix (Speaker-Independent, 9-tone/16-feature)

	1	2	3	4	5	6	7	8	9	%
1	308	0	19	0	0	0	0	0	0	94.2%
2	0	295	0	1	7	0	5	1	7	93.4%
3	12	0	176	0	0	63	0	1	0	69.8%
4	0	0	0	241	0	6	0	0	0	97.6%
5	0	37	0	1	98	0	0	0	0	72.1%
6	0	0	29	7	1	178	0	1	0	82.4%
7	24	0	1	0	0	0	65	4	1	68.4%
8	1	2	23	0	0	6	3	47	14	49.0%
9	0	2	1	16	0	15	0	13	59	55.7%
									AVG	81.9%

Table 4-34 Speaker cs04m Confusion Matrix (Speaker-Independent, 9-tone/16-feature)