2015-01-27

Comparing Human Perception to Computational Classifications of Lexical Tones

Cooper-Leavitt, Jamison

UNIVERSITY OF CALGARY

Comparing Human Perception to Computational Classifications of Lexical Tones

by

Jamison Cooper-Leavitt

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN LINGUISTICS

CALGARY, ALBERTA

JANUARY, 2015

Abstract

This dissertation analyzed the tonal and acoustic properties of utterances produced from five native Thai speakers. The computational model produced classifications based on predictions made by a Hidden Markov Model that simulates tone perception and categorization. The computational model tested the categorization of stimuli taken from both citation and continuous contexts of Thai tonal data, in order to compare the performance of the computational model on both clear and naturalistic stimuli. Two perception experiments were also conducted, involving human listeners, for the purpose of comparing their behavior to that of the computational model. The results reveal that the classifications of lexical tone categories made by the computational model yield some dissimilar learning patterns to that found in human perceptual learning of the same categories.

Acknowledgements

This thesis was a long project that would not have been possible without the guidance of my thesis supervisor, Stephen Winters. He taught me not just about phonetics, but also how to be researcher, scientist and an academic. All these are life-long pursuits and he has set me on a good path in achieving my academic goals. I have also received valuable help and advice from my thesis committee members. Darin Flynn's interest and enthusiasm for language has been an encouragement in pursuing my research in tone languages, phonetics and in all things dealing with language and linguistics. Darin's knowledge of phonology and phonetics has proven to be an asset. From the onset of my research he pointed me to relevant literature that would later become foundational to my own research. Sean Fulop has been a good supporter who is always willing to share his sage advice regarding computational models and linguistic theories. All these individuals have helped me to develop and test the computational models and experiments that are presented in this dissertation.

I would also like to thank my fellow graduate students at the University of Calgary. They have all been good friends and colleagues. I would particularly like to thank Una Chow, Dallas Darie and Mylynn Felt for the help they have provided me in writing and lab help. Also the Higher Clause, Phonetics Club and Verbatim have been great interdepartmental organizations where students and faculty have been able to share thoughts and ideas. Many of these have been sources of inspiration for my own research.

I also must show my gratitude to my parents for the love, support and patience that they have shown to me while I have gone through this process. They have always encouraged me to pursue my dreams and ambitions. Seeing the completion of this dissertation and graduating with my doctorate degree is a fulfilment of their good desires for me.

Lastly, but most especially, I want to thank my loving wife, Pitima, who has been with me throughout this entire process and even long before it. Perhaps more than anyone else, she wants to see me succeed and meet my full potential. She has been my helper and supporter. I also want to express my love to my children, Danai and Nari. My son Danai has been very hopeful for me to finish, and he has shown considerable patience with me especially when dad was too busy for video games or sword fights in the backyard. My youngest Nari has provided me wonderful opportunities for sweet nap breaks when I needed them most. Our faith and conviction in our belief of an eternal loving Heavenly Father has formed the foundation of the strength we possess as a family. This guides me in my pursuit of life, family and career, and ultimately in how I choose to serve those nearest to me.

Dedication

I dedicate this dissertation to my wife. She deserves it more than anyone else I know.

Table of Contents

List of Tables

List of Figures

Chapter One: Introduction

In this dissertation I present my research with two aspects in mind, a practical engineering side in the development and testing of a computational model for tone perception, and a scientific side investigating the implications the computational model holds for the human perception of tone. The computational model developed in this dissertation was a Hidden Markov Model (HMM). I chose a HMM for tone perception, because it is a statistical model that works at a high level of abstraction over a signal (Manning & Schütze, 1999; Rabiner, 1989). In the case of this dissertation, the five lexical Thai tones (High, Mid, Low, Rising and Falling) were the different levels of abstraction over the signal. Fundamental frequency (F0) was the signal.

This dissertation was also a study of the role in which variation in the stimuli and tokens have on the testing and training of a tone perception model. Thus, the topic of this dissertation is on speech perception, and my focus is on tone perception, which is a subset of speech perception.

First on the engineering side, I developed a HMM that was a computational model for tone perception, and I tested how well it worked with Thai tones. Second, I compared the performance of the model with human performance on a similar task to see if the HMM model acted as a legitimate model for human performance. This is a step not commonly taken by engineers when testing their models (Tungthangthum, 1998; Demeechai & Mäkeläinen, 2001).

Wu et al. (2013) compared the accuracy of two tone recognition systems with native Mandarin Chinese listeners. The human participants were tested on tone stimuli given in the context of a vowel, one syllable, two syllables, or three syllables. They also used a neural network classifier system and a HMM system for tone recognition. The neural network system

relied on "phonetic contour features" that were computed every 50 ms over the F0 contour of a segment's duration. The HMMs were computed with F0 in the same manner as the neural network. They found that the tone recognition systems and the humans performed at relatively the same levels of accuracy, but each showed different patterns of errors. Their neural network classifier system performed the least accurate in Mandarin's Dipping and Falling tone categories[1], and their HMM system performed least accurate in the Rising category. The humans performed least accurate in the Dipping and Rising categories. For both recognition systems and for the humans, the High category was recognized with the greatest accuracy out of all the categories. For all humans and for the neural network system, the accuracy rate increased as there were more syllables in the context. The HMM system performed at a high level of accuracy, but included the least amount of context out of the two systems and the human participants. They concluded that with just little context the humans could perform no better than either system. The greatest increases in performance occurred with the human participants, when more context was provided to them.

The main goal of my research was to develop similar experiments as Wu et al. (2013), in developing a HMM computational model for Thai tone perception to compare with human performance in a similar tone perception task. My study was focused on the following questions: Can the HMM correctly identify Thai tones? How well does it correctly identify Thai tones from continuous speech and citation speech? Is there a significant difference between how tones are identified in the two forms of speech? How do human beings actually identify Thai tones?

---

[1] The Mandarin tone categories tested on their tone recognition systems and on the human listeners were High, Rising, Dipping and Falling. For the case of their HMM system they also included the Neutral tone (Wu, Zahorian, & Hu, 2013).

The training and testing paradigms included a pre-test, training and a post-test. The model was run and tested by using several different types of simulations. In two of the simulations the test and training tokens used were taken from either continuous or citation speech. I segmented the continuous tokens from the context of a sentence. These tokens were considered more naturalistic than the tokens produced in a citation context. I expected citation speech to be easier for the model to deal with, but continuous speech is more naturalistic. The practical side of the problem should therefore focus on continuous speech. Continuous speech has more properties found in natural language processing situations than citation speech.

This study also investigated how different levels of quantization of the test and training tokens affected the performance of the model. In the computational model, I use the term quantization to refer to the model's sensitivity to changes in F0 over a sequence of time. For example, the model may be set at a quantization factor of $20\ Hz$. This means that the model only registers changes in F0 for every acoustic change in F0 that is $20\ Hz$ or greater. The level of quantization for which testing and training tokens were analyzed was a parameterized factor in the model. With a more fine-grained quantization of tokens presented to the model, more information from each token could be extracted by the model, but the computing time for the model would increase. This was a practical concern, not a theoretical one, when running the model. However, with a more coarse-grained quantization, less information would be extracted, but the computing time of the model would decrease. The level of quantization of tokens is a practical trade-off in running the model, but there are also theoretically interesting questions to be asked as well. At which level of quantization does the model perform best? At which level of quantization does a model closely resemble human performance?

A tone identification experiment was also conducted with human listeners, who were English speakers with no background or training in tone languages. They were tested and trained on the same tokens used for the tests of the computational model. The tone identification experiment thus acted as an instance of human behavior against which the results of the computational model could be compared. Using naïve speakers only was a departure from several studies on Thai tone perception with human listeners (Laphasradakul, 2010; Teeranon, 2007)[2].

1.1 Using HMMs for a tone perception model

HMMs are useful when equating underlying events with something that is observed (Manning & Schütze, 1999). For a tone categorization model the underlying events may be taken to be the actual F0 of a speech token, that informs the perception of the speech token's lexical tone category. The model components of an HMM can be adjusted to maximize the probability that the model predicts the correct category of an observation (Rabiner, 1989; Manning & Schütze, 1999). *Decoding* is the process by which the HMM predicts an underlying sequence of states leading to an observed event, such as a test token's correct tone category (Rabiner, 1989; Manning & Schütze, 1999). *Expectation Maximization* is a process of maximizing the stochastic properties of the model relative to the fundamental frequency of all tokens used for the training of a specific tone category (Dempster, Laird, & Rubin, 1977). In this dissertation, I applied these processes to five separate sub-models for each of five Thai tone categories.

[2] For the study in this dissertation, gaining access to a large community of native Thai speakers and listeners proved to be problematic.

Speech perception models have benefited from utilizing HMMs. One particular utilization of HMMs is in tone perception models. Tungthangthum (1998) developed a HMM for a Thai tone perception model, which had five sub-models representing each tone category. A native Thai speaker produced ten syllables composed of vowel and tone combinations. Each syllable was produced eight times. The HMM's five sub-models were trained only on the first five sets of syllables that were produced and that were respective to each of the sub-model's specific tone category. Test one used the same test tokens that the model was trained with, and test two used the rest of the tokens that were not used for training. Performance of the model for each test was nearly identical. Tungthangthum's (1998) model only included training and test tokens that were produced in citation form. The model in this dissertation included both citation and continuous tokens for training and testing purposes.

Demeechai and Mäkeläinen (2001) proposed a perception model for Thai tonal syllables produced in a continuous context. Sixteen native Thai speakers read sentences in order to produce tokens for training purposes. Their model contained two components: a tonal HMM system and a syllable HMM system. They tested each system separately using continuous tokens extracted from a Thai corpus, and they also tested the performance of a system that linked the two HMMs into a single system. They referred to this as a linked detection method, and it performed superior to their other models. One particular aspect of these tone perception models is the identification of tone in the context of continuous speech (Demeechai & Mäkeläinen, 2001; Qian & Soong, 2009).

1.2 Thai tone perception in the context of continuous speech

Thai was chosen as the language of investigation in this dissertation for three reasons. First, it is a tonal language that has five distinct lexical tones. They are traditionally referred to

as High, Mid, Low, Falling and Rising tones. Secondly, there is a small but significant native Thai speaker population near the University of Calgary who were available to participate in this research. Thirdly, I am familiar with the language and the community, which helped make it easier to generate the database of spoken words for testing both the model and the naïve listeners.

Another compelling reason to investigate Thai is the unique nature of its level and contour tones. An examination of pitch and duration in both citation and continuous speech show that the Thai tonal system is complex (Morén & Zsiga, 2006). Thai level tones are less static than their description may imply. This is particularly true for the High tone category (Teeranon, 2007; Zsiga, 2008). Figure 1.1 and Figure 1.2 show the F0 contours of the five Thai tones. Figure 1.1 shows these tones produced in a citation context, and Figure 1.2 shows these tones produced in a continuous context. These tones were produced by the same speaker using the same words uttered in both types of contexts. The details on how these utterances were produced is found in Chapter Two. The vertical axis in these figures is frequency measured in Hertz.

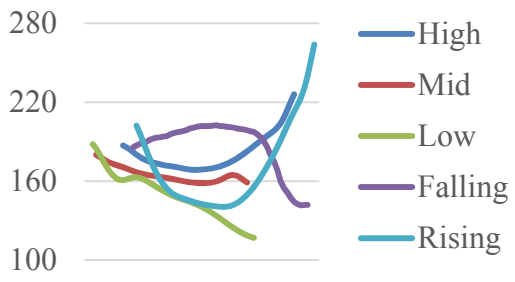

Figure 1.1 Citation tones

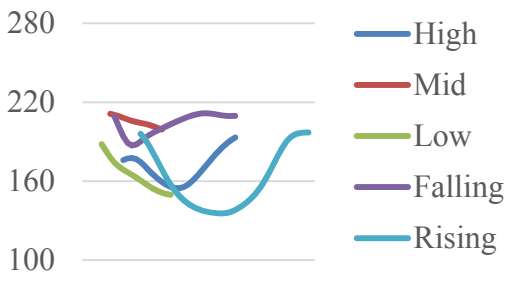

Figure 1.2 Continuous tones

Thai tones spoken in a continuous context show less of a contour and a shorter duration (Zsiga & Nitisaroj, 2007; Nitisaroj, 2006; Potisuk, Gandour, & Harper, 1997). The Thai tone system lends

itself well to a study of tones spoken in a continuous context, and to the effect this context has on tone perception, in both a computational model and in a tone perception experiment with human listeners.

Some early studies on Thai tone perception investigated the role of an utterance's F0 in perception. Abramson (1986) developed four experiments investigating F0 and the perception of pitch in the Thai tonal system. On the basis of these experiments he argued for the primacy of the F0 contour as the carrier of tonal information in Thai. This is expressly evident with Rising and Falling, where F0 contour acts as a strong cue for the correct tone category.

Other Thai tone perception experiments have investigated the role of F0 along with other tone features. Wayland and Guion (2003) performed a Thai tone identification study with three groups of listeners. One group was composed of native Thai listeners, and the other two were native English listeners, where one of these groups already had some L2 training in Thai. Each group was asked to discriminate between low and mid Thai tones in syllables that were closed, those that ended with a consonant or sonorant, and in syllables ended with just a vowel. Their results showed that all groups found the open syllables to be more difficult than the closed syllables. Ramadoss (2011) argues that tone perception is not only affected by fundamental frequency and the pitch contour of tones, but it is also due to features at the segmental level of the syllable such as open or closed syllables and syllable length.

Other studies have investigated the role that a listener's native language plays in tone perception. Wayland and Li (2008) tested the ability of native English listeners who had no prior knowledge of Thai or any other tone language to discriminate between Thai low and mid tones. They also tested a group of native Chinese listeners on the same test. The native Chinese group performed better than the native English group, but both groups showed a considerable amount

of improvement between the pre-test and post-test. They concluded that prior knowledge of a tone language helped the native Chinese discriminate better between the Thai Low and Mid tones. Similar results were also shown by Francis et al. (2008), who found that native Mandarin listeners performed better at identifying Cantonese tones than native English speakers. Their study also showed that language background had a significant influence on which type of tone feature listeners would focus on. English listeners focused on the F0 pitch level of the Cantonese test stimuli during the categorization task, while Mandarin listeners were more focused on the pitch contour of the test stimuli during the same task.

Studies have also been conducted to investigate the role that the speech type used for testing and training stimuli play in tone identification tasks. Laphasradakul (2010) studied the effects the continuous Thai stimuli had on one group and that citation stimuli had on another. Her results showed that all groups performed significantly better on citation stimuli than on continuous stimuli.

A recent study by Schaefer and Darcey (2014) investigated naïve listeners' L1 perception of pitch and its effect on the listeners' ability to distinguish Thai tones. Four listener groups were included in their study. They included a Mandarin group (lexical tone), a Japanese group (lexical pitch accent), an English group (lexical stress), and a Korean group (no lexically contrastive pitch). Participants were tested on an AXB tone categorization test, where they were given AB pairs of tones, and a test stimulus, X, and they were tasked to identify whether X's tone matched with A's tone or B's tone. They used real words from all five Thai tone categories. The Mandarin group of participants outperformed all the other groups, and the Japanese group outperformed the English and Korean groups. They concluded that English and Korean listeners found the categorization task to be more difficult due to the lack of lexically contrastive pitch

elements in these groups' respective native language. In my study, I also tested naïve English speakers in their perception of Thai tones. The inherent difficulty in categorizing Thai tones for English listeners was considered to be comparable to an equally uninformed computational model, which was also tasked to identify Thai tone categories.

1.3 Testing paradigm

In my dissertation, the methodology used to test both the tone perception model and the human listeners in the tone identification experiment includes three phases: a pre-test, training, and a post-test. Test results are analyzed between pre-tests and post-tests to determine if training had a significant improvement on performance on the tone identification task in the post-test, i.e. if perceptual learning had occurred. A similar testing methodology was used in Laphasradakul (2010). Another factor to be tested was the role that the production context of the testing and training tokens and stimuli would play between the pre-test and post-test. Both citation and continuous testing and training stimuli were used in the listening experiment, and the testing and training tokens used in the computational simulations were also from both a citation and continuous context.

The testing paradigm used for the perception experiment and the computational simulations were designed to be as similar to each other as possible. For example, the computational model cannot recognize an *ad hoc* relationship between category label names and the actual tone categories, such as "high" and its appropriate F0 contour. These labels were thus not used in the tone identification experiment, so as to not give the human subjects an extra advantage during the tone identification task that the HMM did not have.

1.4 Framework of this dissertation

There are two main components to this dissertation. The first is a description of the design and testing of the tone perception model, and the second involves a tone identification experiment. In chapter two I describe the Thai language and its tone system, and how it was used to create a database for the tone perception model and also for the tone identification experiment. I give a statistical description and an analysis of the Thai database used in the model and the experiment. In chapter three, I discuss how I apply a HMM and its components to tone perception, and I discuss two methods of analyzing the model: *Decoding* and *Expectation Maximization (EM)*. Decoding is used to predict the probability that correct tone category will be produced by the model, and EM is used to train the model in order to improve its predictive properties. In chapter four, I discuss the framework used to test the tone perception model, the different experiments that were run on the model, and the results of those experiments. Chapter five discusses the tone identification experiment, the listeners who participated in the experiment, the testing methodology, and how the results between tests and groups were analyzed. Chapter six concludes the dissertation with a discussion of the theoretical implications of the model and experimental results. In particular I discuss how the results relate to Exemplar Theory.

Chapter Two: Creating the Thai Datasets

In this chapter I discuss some of the phonetic qualities of Thai lexical tones that were considered when making the database of Thai stimuli for the computational simulations and the perception experiment. The phonetic quality that mattered most for this research was the acoustic features of Thai tones[3]. The two contexts that are investigated here are: (1) citation form and (2) continuous speech.

2.1 Phonetic description of Thai tones

Lexical tones are generally considered to be a suprasegmental feature (Lehiste, 1970). Tones are not a segment but form part of an entire syllable, and they have a semantically significant F0 contour which can distinguish two otherwise identical syllables (Abramson, 1960; Pike, 1957). In Thai there are specifically five lexically contrastive tones: high, mid, low, falling, and rising. The 'high' designation indicates a higher F0. A low tone indicates a lower F0, and a mid tone indicates an F0 that is in between these two. Falling and rising tones indicate a change in the F0 over the entire syllable. Falling starts with a higher F0 and ends with a lower F0, and rising starts with a lower F0 and ends with a higher F0 at the end of the syllable.

Syllable length and phonetic segments are also features that phonologically interact with the lexical tones (Yip, 2002). Thai has a contrast between long and short vowels (Gandour, 1975). Sonorant-final syllables can have any of the five tones. Stop-final syllables with long vowels may only have low and falling tones, and stop-final syllables with short vowels may only have low and high tones. The computational model focuses on identifying lexical tone from F0

[3] Appendix A includes a Thai IPA chart and a short description of Thai phonetics.

contours only, and not from the syllable length or structure. It is quite possible that syllable structure and length play a role in lexical tone perception and categorization, and could be factored into future models and experiments.

The tones in Table 2.1 provide examples of each of the five lexically contrastive Thai tones on the syllable /kʰa:/ (Nitisaroj, 2006). Simply by changing the level of F0 on the syllable, the tone and the meaning of the word changes.

Tone	Example	
Falling	/kʰâ:/	value
High	/kʰá:/	to trade
Low	/kʰà:/	galangal (a spice)
Mid	/kʰa:/	to be stuck
Rising	/kʰǎ:/	leg

Table 2.1 Thai lexical tones

Pike (1957) and Abramson (1960) give a phonological description of the five Thai tones. They use two tone targets, H and L, to capture the five lexical distinctions shown in Table 2.1. H represents a high pitch and L a low one. Thus, for the lexical tones, a high tone is represented as (H), a low tone is (L), and the contour tones are represented as falling (HL) and rising (LH). The mid tone is considered to be a "neutral" tone in this system, so a word that has a mid tone does not have one of the tone features (H or L) associated with its syllable. Hence the blank representation in the orthography (see Table 2.1).

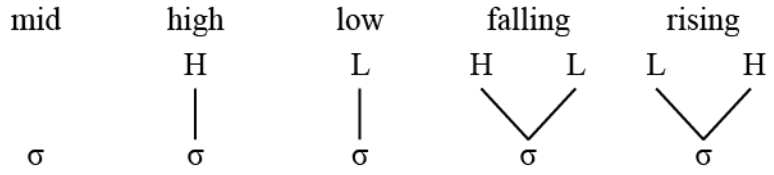

Figure 2.1 Segmental representation

If the tone contour of a word is falling, it is represented with a feature H and a feature L, in that order, associated with the word's syllable. Likewise, if a word has a tone contour that is rising, it is represented in the reverse manner with a feature L followed by H, and both features are also associated with the word's syllable. However, this autosegmental representation oversimplifies the actual acoustic properties of the lexical tones. Such phonological descriptions of lexical tones are not very useful for developing a computational model of tone identification from actual spoken Thai words. This very abstract way of looking at the tone structure glosses over all sorts of variation and phonetic detail in the actual production of each tone. The aim of the computational model and the tone categorization task is to dig into the phonetic details and come up with the same meaningful classifications as in Table 2.1.

2.1.1 Acoustic representations of lexical tones

The F0 of a word varies quite dramatically depending on the context and speaker. Potisuk, Gandour and Harper's (1997) analysis of Thai shows that each of the tone categories has a contour shape, instead of some having level F0 throughout the syllable segment. Rising and falling contours exhibit less of a change in pitch when spoken in a sentence or utterance than when spoken in isolation as a citation form (Potisuk, Gandour, & Harper, 1997). To illustrate this difference in contour between the two contexts, Figure 1.1 shows the F0 of all five tones produced in a citation context from one of the speakers in this study. Figure 1.2 shows the same five tones produced from the same speaker produced in a continuous context. The Rising tone in the citation context has a much greater range of frequency than in the continuous context. The Falling tone in the continuous context is much shorter in duration, it is clipped, and it has a much shorter range in frequency.

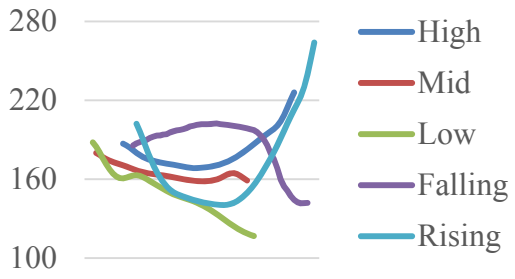

Figure 2.2 Citation tones

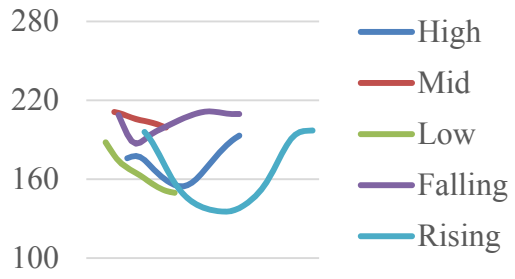

Figure 2.3 Continuous tones

Figure 2.4 shows the pitch contour for *baan,* 'dull', which has a falling tone. This word is uttered in a citation context. It has a falling tone showing a rise followed by a dramatic lowering in pitch. The initial stretch of disconnected F0 comes from the [b] in the utterance, which may also explain the initial rise for the "falling" tone. This is contrasted with the pitch contour in Figure 2.5 of the same word which was uttered by the same speaker in continuous speech. The rise in pitch is less pronounced and it is not followed by a lowering in pitch at all, which is a surprising contrast due to the fact that it is a "falling" tone.

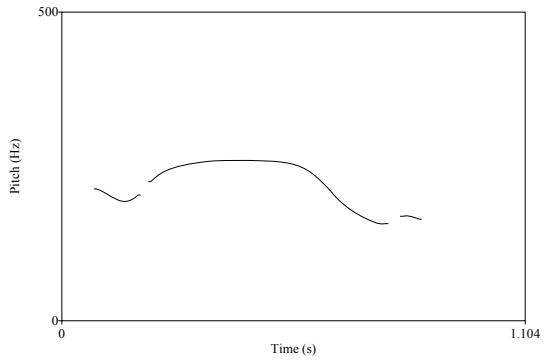

Figure 2.4 Pitch contour of falling tone spoken in citation

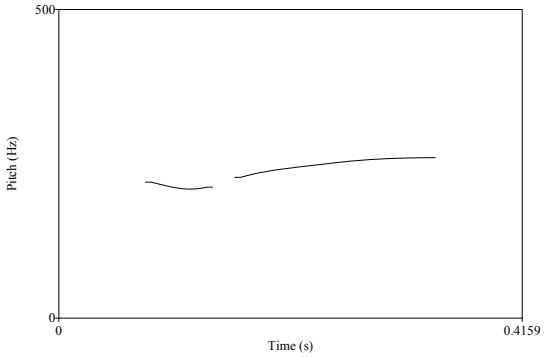

Figure 2.5 Pitch contour of falling tone spoken in continuous speech

2.2 The Datasets

As mentioned in chapter one, one aim of this research was to test how well the listeners and the model can process both kinds of speech, with the recognition that the tones in continuous speech should be more difficult to identify. However, continuous speech is what human listeners have to deal with when processing natural language, and thus my aim was to makes sure that the computational model could also process the same lexical tone data as well. To test this research goal I created two datasets. The first dataset contained word tokens that were recorded in citation form. The second set contained word tokens that were recorded from continuous speech (single sentence utterances). These tokens are more realistic representations of natural language. The tokens in the latter dataset were taken from a collection of recorded sentences, where each sentence contained a stimulus word that was later cut out of this context and then used as a stimulus token for this dataset.

2.2.1 Methods and materials

Five native female Thai speakers were recruited from Calgary, and each of them produced 762 citation tokens[4] and 20 continuous tokens. Only female speakers were recruited for this study in order to somewhat standardize the range in frequency between different speaker's voices. The speakers were between the ages of 25 to 36 years, were born and raised in Thailand, and each of them spoke English as their second language. They all used Thai on regular, daily basis. Three of the speakers were from the eastern provinces in Thailand, one was from southern Thailand, and one was from central Thailand. All the speakers have lived in Canada for the past two to five years, and they have spoken English for two to twenty years. One speaker was learning English; three speakers considered themselves to be fluent English speakers; one speaker considered herself to have a near-native level of English fluency. Two of the speakers had earned doctorate level degrees from universities in North America, one was currently attending graduate school, and two others had received bachelor's degrees from universities in Thailand.

2.2.1.1 Citation database

Table 2.2 shows five example tokens from the production list that were used for the citation stimuli. Each speaker was recorded individually in a sound-attenuated booth at the University of Calgary's phonetics laboratory. These tokens were originally recorded for a

[4] The website, http://www.thai-language.com/dict/ was the reference source for the tokens used to create the database. Some of the citation word tokens were accidently missed during some of the speaker's recording sessions. Speakers f6 and f10 recorded 761 words that were used as tokens, and all other speakers recorded 760 words. Also, due to some minor irregularities while recording speakers, some utterances were not suitable to be used as tokens. Speaker f6 and f10 missed one, and speakers f7, f8 and f9 missed two tokens.

different project, which is why so many items were recorded (Winters, 2010). However, four of the five native female Thai speakers recorded here were new speakers recruited for this project.

Item #	Token Name[5]	Orthography	Tone	Lexical Category	Definition
101	bpehk	เป็ก	H	noun	thumbtack
177	dtaa	ตา	M	noun	eye
3	bpraa	ปร่า	L	adjective	unpalatable
222	dtaaem	แต้ม	F	noun	point
357	duuhr	เด้อ	R	modifier	clumsy

Table 2.2 Example Tokens and definitions

The speakers read a Thai word which was presented on a computer screen for three seconds. Each word was presented in a standard 60pt, Thai font. There was no extra graphical indication of the intended tone or pronunciation given. The words were presented to the speakers using a PowerPoint presentation. Speakers sat 1.5 meters from the computer screen, and were directly in front of a microphone that had a pop-filter attached to it. Words were presented in 50 word blocks, each of which took approximately 150 seconds to produce. Additional time was used to re-record any mistakes the speakers felt they had made after each 50 word block. At the interval between each block, speakers took a short break and a drink of water to refresh their voices. Each block of 50 utterances was saved as an individual sound file, which was then post-processed in the lab using Praat. Each sound file was segmented into 50 individual word sound files. The full set of words is provided in Appendix B.

[5] The transcription method used for each token came from the website, http://www.thai-language.com/dict/. The website did not use standard IPA conventions for their transcriptions. For example they used **bp** to transcribe the voiced unaspirated stop /b/. I chose to follow their transcription method, since it facilitated a standard keyboard. This made it easier to write the Perl, Praat and MatLab scripts. These scripts have been included in Appendix D.

2.2.1.2 Continuous database

Each of the speakers also produced twenty Thai sentences with individual stimulus words, from the citation database, spoken in the context of a sentence. The speakers read twenty sentences, taking a short break in between each sentence. The speakers were not told which word was the target word in each sentence. The sentences were written in the same Thai font used in the citation form tokens, and they were read from a sheet of paper.

1.	แดง	อยาก	กิน	หมู		ปิ้ง		
	Deng	yaak	kin	muu		**bping.F**		
	Deng	want	eat	pork		grilled		
	"Deng wants to eat grilled pork."							
7.	ดา	โทรไป	จอง	โต๊ะ	กิน	ข้าว	มื้อเย็น	
	Daa	thow bpai	jaawng	**dto.H**	kin	khau	mua yen	
	Daa	telephone	reserve	table	eat	food	dinner time	
	"Daa phoned to reserve a dinner table for dinner time."							
12.	หน่อย	ชอบ	ทาน	ข้าวต้ม	ใส่	ตับ	หมู	
	Noi	chaawp	thaan	khau dtom	sai	**dtap.L**	muu	
	Noi	likes	eat	rice porridge	with	liver	pork	
	"Noi likes to eat rice porridge with pork liver."							
16.	เอ	วิ่ง	ไล่	ปู	ลม	บน	หาด	ทราย
	Aey	wing	lai	**bpuu.M**	lowm	bowb	haad	sai
	Aey	run	chase	crab	wind	on	beach	sand
	"Aey chased the crab on the windy beach."							
21.	ชาว	นา	ช่วย	กัน	ถอน	ต้น	หญ้า	
	chaow	naa	chuay	kan	**thaawn.R**	dtown	naa	
	farmer	field	help	each	pull	plant	weed	
	"Farmers help each other pull weeds."							

Table 2.3 Glosses and translations of sample Thai sentences

The sentences shown in Table 2.3 include glosses and translations of some sample Thai sentences presented to the speakers. In these samples, the first line is the Thai script segmented into individual words. The second line shows a traditional orthographic transcription of each word. The target word that was used as a test token is bolded and its lexical tone category is indicated after the transcription. In these three samples each stimulus token happens to be

falling. The third line is a word by word translation[6]. The fourth line is an approximate English translation of the Thai sentence. There were five stimulus words for each tone category, for a total of 25 sentences recorded for each speaker. The stimuli were distributed as one utterance-initial, 13 utterance-medial, and 11 utterance-final. The corresponding words in the continuous stimuli were also segmented from the sentences they were produced in, by using Praat, and were stored in individual sound files. See Appendix B for the full list of Thai sentences along with their glosses and translations.

2.2.2 Statistical description of the database

I chose to do an acoustic analysis of this data for two reasons. First, I wanted to show the variability between speakers that the model (and listeners) had to deal with. Second, I wanted to find out what acoustic differences there were between each of the speakers.

Autocorrelation pitch tracking was used in Praat to measure the fundamental frequency (F0) of each token in 0.01 second intervals. The pitch data was stored in matrix files. Table 2.4 shows the mean pitch and standard deviation for each speaker.

Speaker[7]	*Mean* (Hz)	*SD* (Hz)
f6	166.95	20.24
f7	205.09	28.1
f8	178.31	22.63
f9	235.89	30.52
f10	229.67	32.67

Table 2.4 *Mean(SD)* frequency *(Hz)* per speaker

[6] CLF stands for the grammatical category of classifier.
[7] Since these speakers were added to a pre-existing database, the numbering began at six instead of at one.

The next two tables give a breakdown of descriptive data for each lexical tone category, by speaker. Table 2.5 shows the mean frequency and standard deviation per lexical tone category. Table 2.6 shows the minimum and maximum ranges of frequency for each speaker's lexical tone category. Figure 2.6 illustrates the distribution of the total frequency range of all falling tones for each speaker, as an example of the cross-speaker variability that the model (and listeners) had to contend with during the tone identification task.

	Falling	Low	High	Medium	Rising
f6	187.07 (19.89)	150.85 (14.67)	177.62 (13.61)	163.55 (8.78)	163.88 (24.89)
f7	224.13 (40.66)	189.31 (18.52)	226.73 (29.62)	203.92 (13.27)	201.20 (23.92)
f8	197.96 (24.93)	165.91 (17.26)	200.28 (15.75)	171.14 (13.70)	177.19 (22.03)
f9	261.71 (38.20)	220.98 (17.67)	254.15 (28.88)	229.82 (21.38)	228.67 (29.38)
f10	260.71 (40.64)	211.30 (23.05)	250.45 (31.25)	223.34 (16.74)	219.51 (32.05)

Table 2.5 Speaker *Mean(SD)* frequency *(Hz)* per tone category

	Falling	Low	High	Medium	Rising
f6	131.08:264.36	108.77:323.11	153.76:285.15	142.99:216.14	131.93:246.61
f7	146.19:337.96	83.99:286.02	154.53:344.94	159.63:309.39	150.03:333.34
f8	138.82:337.96	130.18:222.15	151.08:260.71	134.67:238.42	143.80:262.03
f9	163.99:344.79	171.23:319.59	178.59:347.25	169.39:328.43	187.38:340.65
f10	104.71:340.88	103.36:346.26	103.36:346.26	175.74:324.60	104.94:337.53

Table 2.6 Speaker Range *(Min:Max)* frequency *(Hz)* per tone category

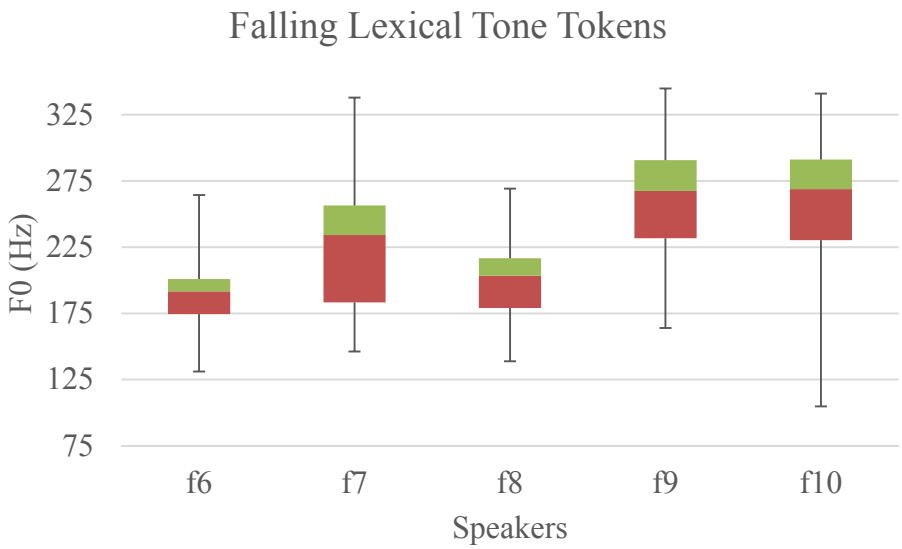

Figure 2.6 Distribution of citation frequencies for all falling lexical tones per speaker

Even though the boxplot in Figure 2.6 shows the distribution of the mean F0 frequency for each speaker and category, the figure does not contain a time element. Figure 2.7 is a plot of averaged F0 values for each speaker's set of high tone productions, that was sequenced into time points for every tenth portion of the token. The mean F0 frequency for each of the portions was calculated and shown as a sequence in Figure 2.7, and for the other lexical tone categories all the way through to Figure 2.11. These plots show the mean frequency level and also the degree of contour for each speaker and each tone category. There is some contour in the supposedly level tone categories, but the contour is not nearly as dramatic as seen in the rising and falling tones.

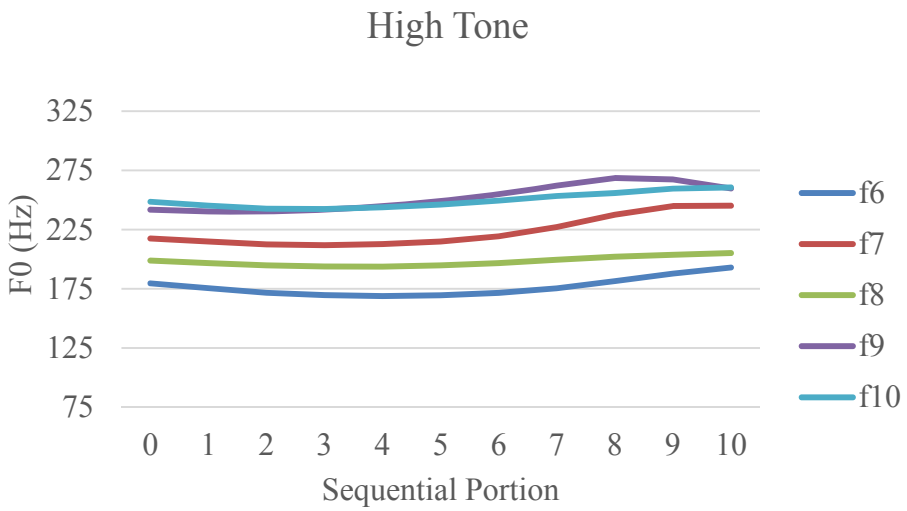

Figure 2.7 Temporally proportioned sequence for high tones

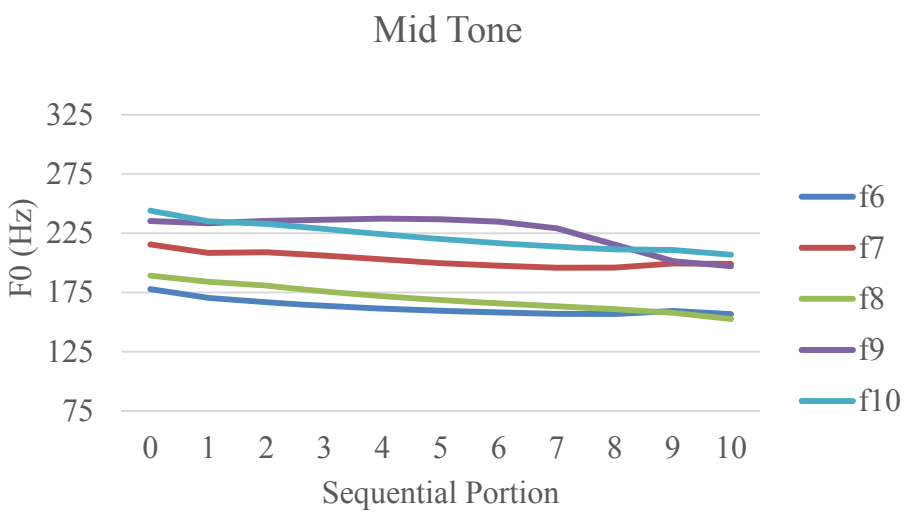

Figure 2.8 Temporally proportioned sequence for Mid tones

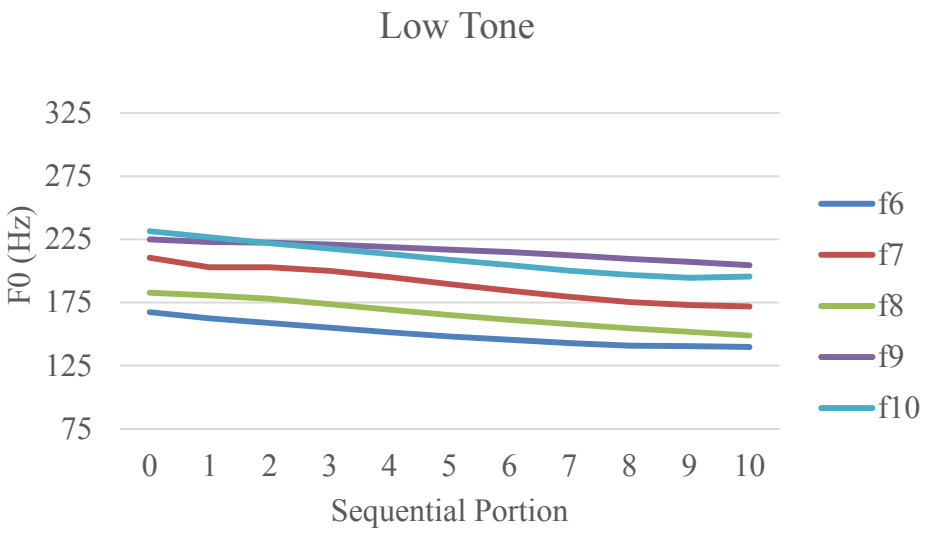

Figure 2.9 Temporally proportioned sequence for Low tones

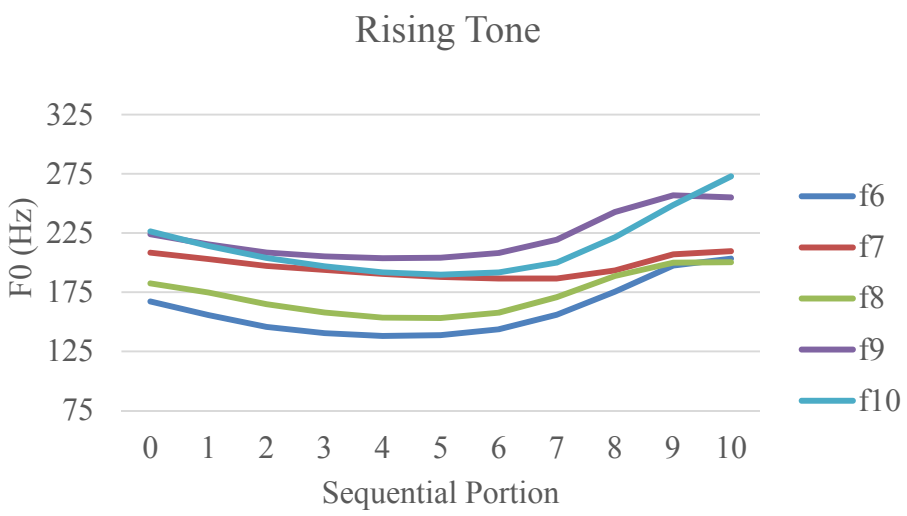

Figure 2.10 Temporally proportioned sequence for Rising tones

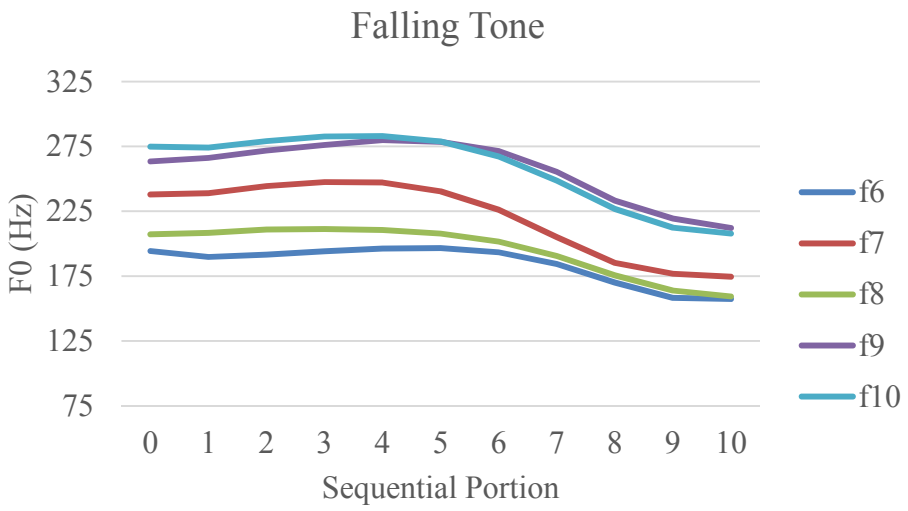

Figure 2.11 Temporally proportioned sequence for Rising tones

A two-way ANOVA was run to test the effects of tone and speaker on mean F0 frequency. Speakers was treated as a between-subjects factor, and tone was treated as a within-subjects factor. There was a significant effect of speaker, $F(4,241) = 130.61$, $p < 0.001$, and there was also a significant effect of tone, $F(4,241) = 96.54$, $p < 0.001$. There was also a significant interaction of speaker and tone on mean frequency, $F(16,241) = 1.72$, $p = 0.043$. These results show that significant differences exist between the different speakers and tones, and the interaction of the two factors also shows that speakers within each tone category were also significantly different.

2.2.3 Description of Continuous Tokens

Table 2.7 shows the mean and standard deviation of F0 frequency for tones produced in the continuous context by each speaker and per each category. Table 2.8 shows the range in terms of the minimum and maximum frequency for each speaker and category.

	Falling	Low	High	Medium	Rising
f6	201.70 (13.84)	152.08 (9.20)	173.15 (11.70)	173.10 (18.68)	159.59 (21.04)
f7	241.28 (27.92)	176.64 (13.24)	207.96 (17.23)	201.43 (17.26)	194.27 (30.55)
f8	212.43 (28.69)	157.57 (15.89)	183.04 (13.66)	169.99 (16.81)	164.83 (15.54)
f9	298.47 (35.99)	211.17 (17.15)	258.36 (22.92)	261.38 (35.37)	232.89 (29.58)
f10	258.45 (19.99)	205.27 (16.72)	218.39 (14.77)	221.95 (22.13)	207.66 (24.01)

Table 2.7 Speaker *Mean(SD)* frequency *(Hz)* per tone category

	Falling	Low	High	Medium	Rising
f6	163.78:228.68	130.08:167.47	154.59:194.63	142.99:211.15	131.93:197.72
f7	160.38:288.12	155.01:196.96	175.88:246.46	165.46:239.17	153.43:325.95
f8	158.35:269.17	136.35:205.37	165.48:221.66	143.40:209.16	143.80:191.86
f9	204.89:344.79	176.27:254.58	213.89:297.78	192.57:328.43	193.71:314.32
f10	188.83:288.35	176.46:251.98	196.56:266.16	188.62:268.01	183.71:310.66

Table 2.8 Speaker Range *(Min:Max)* frequency *(Hz)* per tone category

Figure 2.12 shows the distribution of total frequency range of the continuous falling tone for each speaker. Compared to Figure 2.6, the range is quite constrained for the continuous falling tone. Figure 2.13 through Figure 2.17 are graphs showing the mean F0 sequential portion for tone category and speaker.

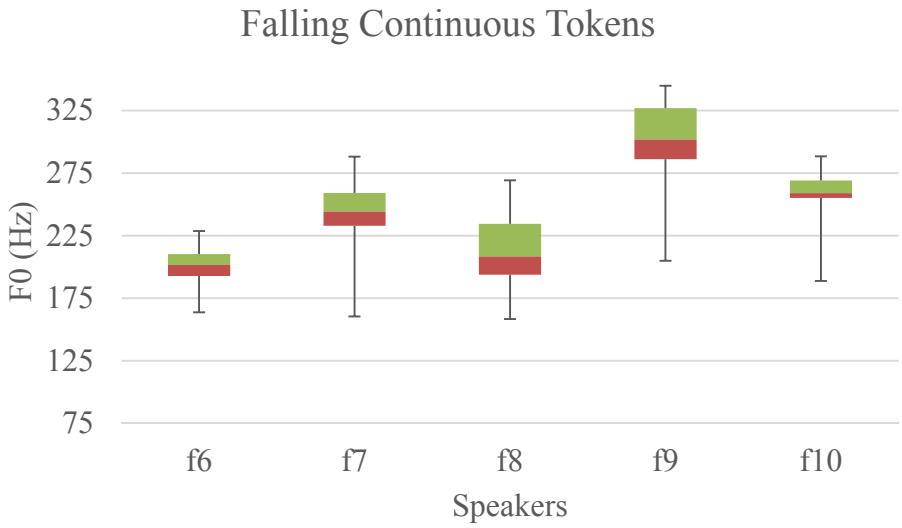

Figure 2.12 Distribution of continuous frequencies for all falling lexical tones per speaker

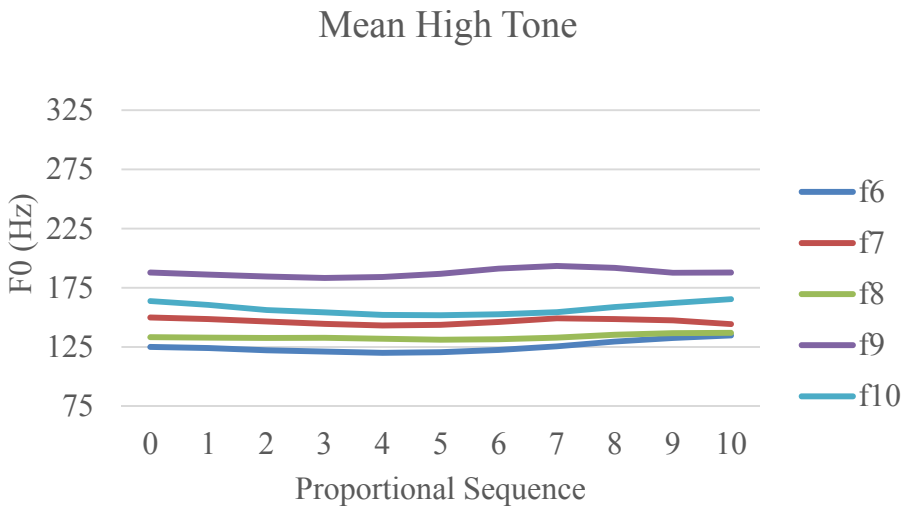

Figure 2.13 Sequence for High tones

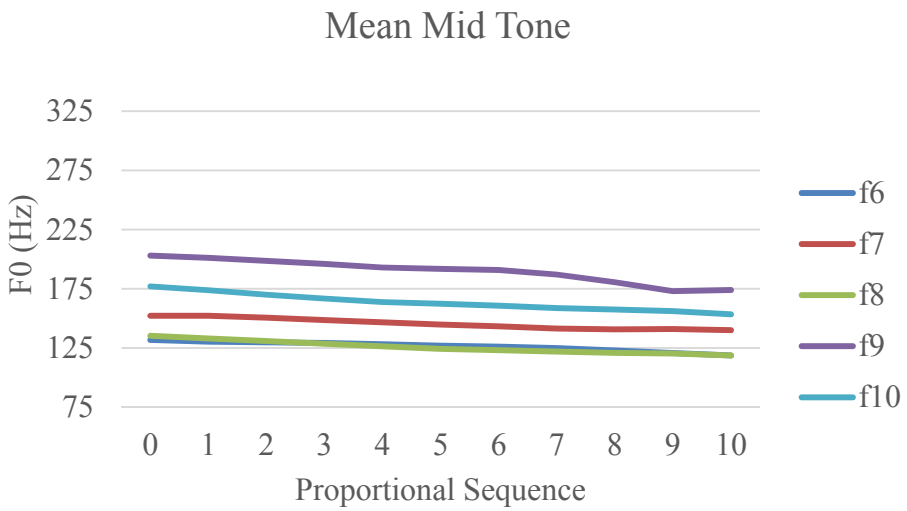

Figure 2.14 Sequence for Mid tones

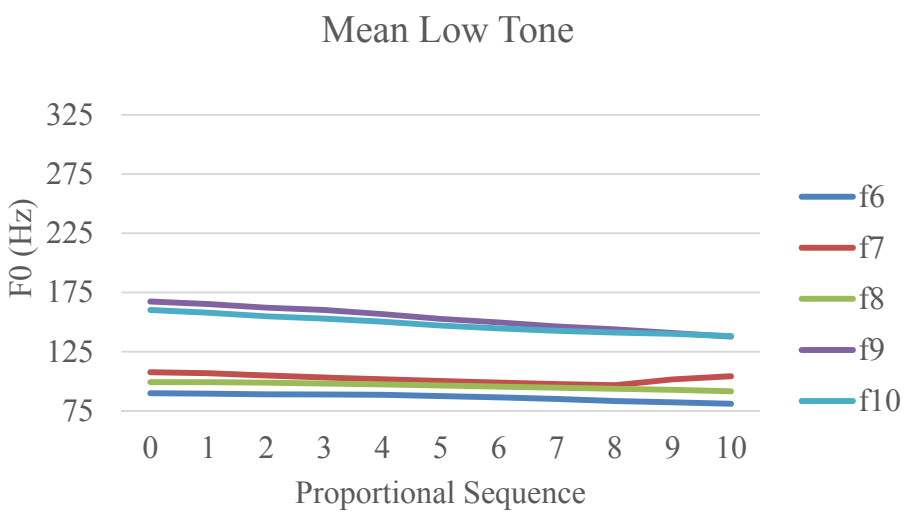

Figure 2.15 Sequence for Low tones

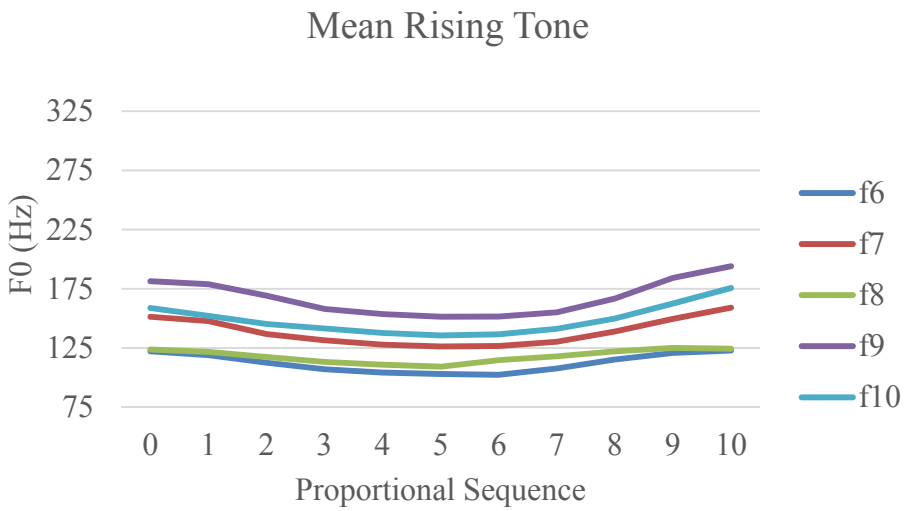

Figure 2.16 Sequence for Rising tones

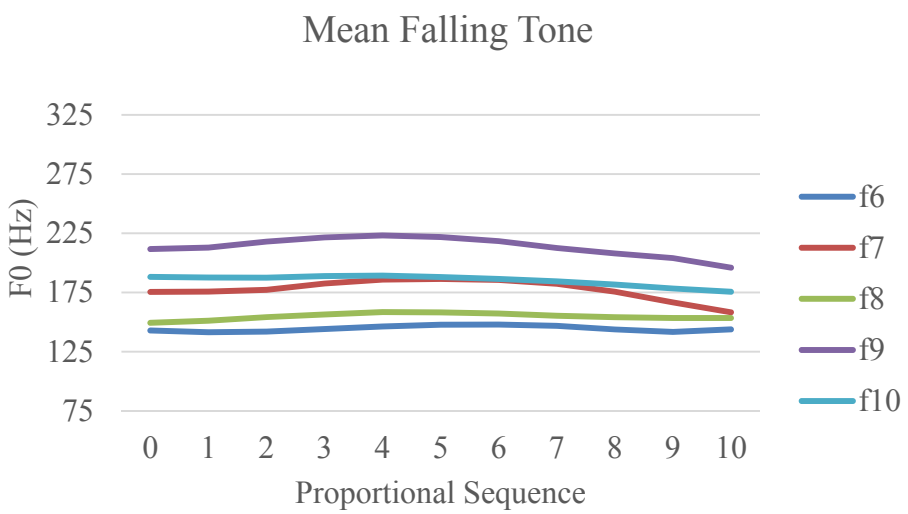

Figure 2.17 Sequence for Falling tones

A 2-way ANOVA was also run to test the effect of speaker and tone on the mean F0 of the continuous tokens. Speaker was treated as the between-subjects factor, and tone was the within-subjects factor. Speaker had a significant effect on the mean F0 frequency, $F(4,241) = 351.9$, $p < 0.001$, and tone category also had a significant effect, $F(4,241) = 196.7$, $p < 0.001$.

There was also a significant interaction of speaker and tone category, $F(16,241) = 13.1$, $p < 0.001$. Within the set of continuous tokens, the speakers and the tone categories were significantly different from each other in terms of F0 frequency.

I ran a 2-way ANOVA to test the effects of speaker and token type on the mean F0. For the purpose of this test, only those citation tokens that had an analogous continuous token were included in the citation list of tokens. This was done to ensure that the tests were comparing the exact same words, and that the only difference was in production context. Token type was the within-subject factor, and speaker was the between-subjects factor. Token type did not have a significant effect, $F(1,538) = 0.83$, $p = 0.362$, but speaker had a significant effect, $F(4,538) = 167.20$, $p < 0.001$. The interaction between token type and speakers had a significant effect, $F(4,538) = 2.54$, p 0.039. Table 2.9 shows the mean frequency differences between the citation and continuous tokens. Speakers f7, f9 and f10 consistently had a lower mean F0 in continuous speech than in citation form[8].

Speaker	Citation (*Hz*)	Continuous (*Hz*)	$t(54) =$	p
f6	118.77	118.92	-0.12	= 0.907
f7	150.30	142.01	3.19	= 0.002
f8	126.76	125.56	0.71	= 0.480
f9	176.10	182.30	-3.13	= 0.003
f10	169.30	160.60	3.73	< 0.001

Table 2.9 Mean frequency differences between token types

[8] The two token types were significantly different from each other. However, it is important to point out that the listeners in the perception experiment are not only sensitive to F0, but they may be sensitive to other acoustic information present in the speech signal, such as vowel formant frequency, amplitude, duration, etc. This will be discussed later in the dissertation.

2.2.4 Single speaker database

Natural language is of course not just limited to one speaker, but to multiple speakers. However, I wanted to create conditions in which the computational model would operate at its best performance, and in which I could make sure that it worked in principle. The intent for single speaker database was to create a set of conditions where the computational model could exhibit its best performance due to using a simplified set of tokens. One way I considered doing this was to create a simplified database with only a single speaker for the model to be tested and trained on. By choosing a limited set of tokens from just one speaker, the amount of variation introduced into the model would be minimal compared to using all of the tokens from all speakers. A simpler database would have less variation in the type of tokens used for testing and training the complete database with all five speakers and all their tokens. This database included the productions from just a single speaker, and only 35 words for each tone category. This database served the engineering side of testing the computational model, and provided a benchmark of sorts for the computational model when it was run on a dataset with more speakers.

I wanted to target the tokens from the speaker that exhibited the least amount of variability within each category of her tokens. I adapted Lobanov's (1971) method for normalizing vowel formants from multiple speakers to a method of normalizing the F0 of tones. The mean F0 frequency and standard deviation values that were previously calculated and shown in Table 2.4 were used to calculate a normalized score for every token produced by each speaker. The normalized score was derived for each token by taking the F0 frequency from the beginning,

middle[9] and final time sequence of the token. A z-score for each of the three F0 frequencies was calculated (using the values in Table 2.4 as the mean and standard deviation for each speaker) and converted to an absolute value, and then the sum of these three absolute values was the score for the token. The absolute value of all the z-scores gave an overall measure of how much variability in F0 there was in each token.

The entire set of tokens for each speaker was assigned a score in this manner, and then sorted by score from least to greatest per each speaker. The speaker with the lowest median score was taken to be the speaker with the least variation, and was chosen as the speaker for the simplified single speaker database. This speaker was f6. Next, the tokens were sorted from least to greatest by lexical tone category and score. For each tone category, the first 35 tokens with the least variation were chosen to be included in the simplified database.

2.2.5 Statistical Description of the Simplified Dataset

Speaker f6's recordings were used for this dataset. The list of tokens used to create the dataset is found in Appendix C. Table 2.10 shows the descriptive statistics for each category in the single speaker database.

Category	Min:Max	Mean (SD)
High	158.06:233.55	176.94 (10.05)
Mid	150.10:197.32	164.95 (7.33)
Low	130.67:195.65	152.93 (12.55)
Rising	134.94:235.38	163.58 (23.07)
Falling	137.12:264.36	186.35 (16.19)

Table 2.10 *Min:Max* and *Mean (SD)* frequency *(Hz)* for single speaker database

[9] This is at a 50% of a word's entire duration.

The distribution of F0 frequencies for all of the tokens used in the single speaker database is shown in the boxplots in Figure 2.18. Figure 2.19 shows the sequential portions of F0 for each tone category.

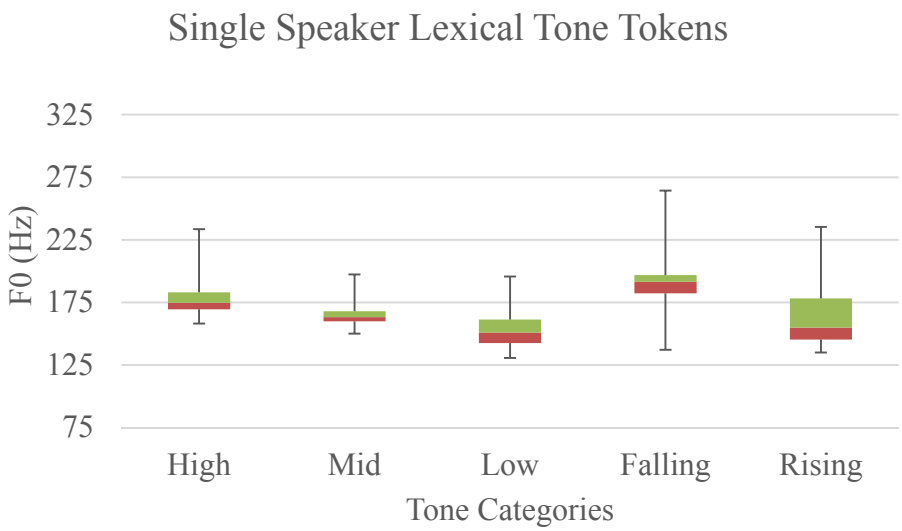

Figure 2.18 Distribution of F0 frequency for Single speaker database

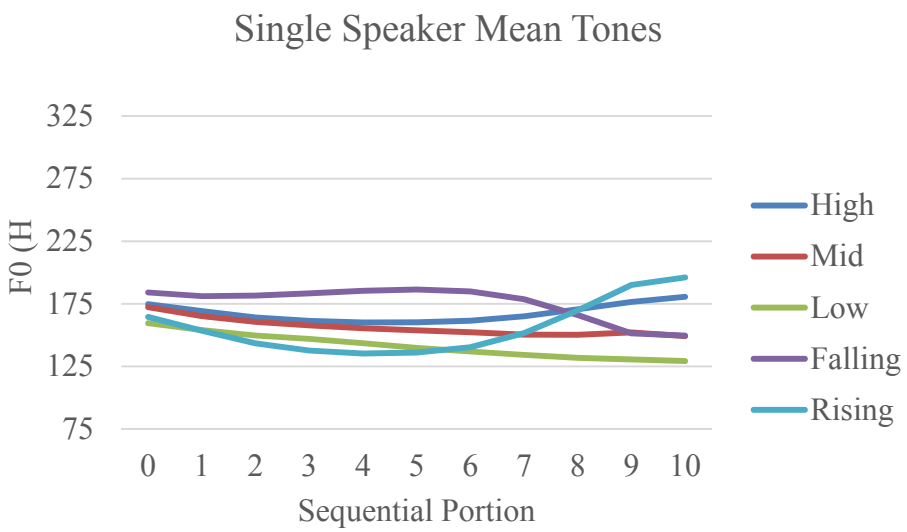

Figure 2.19 Sequential portions for single speaker database

Chapter Three: Applying a Hidden Markov Model to Tone Perception

Any system or phenomenon that has a probabilistic outcome where not all the factors of the system are observable can be modeled as a Hidden Markov Model (HMM). HMMs have been applied to many different phenomena including problems and topics in genetics, finance, facial recognition, gait recognition and speech perception. A HMM is a system with a defined set of states that obey stochastic constraints (Baum & Petrie, 1966; Baum, Petrie, Soules, & Weiss, 1970). Each state in the model is a hidden variable. Observations are sequences of symbols linked to the hidden states. Over time, the hidden state of the model may change. One of the goals of a HMM is to determine the probability that an observation may be produced by all the possible state sequences in the model. The sum of the probability of all of these state sequences represents the probability of an observation, given the model. This conditional probability is what makes the system Bayesian.

Seminal discussions of HMMs and their application to speech perception can be found in Rabiner (1989) and Manning & Schütze (1999). From these sources, I present the main components of a HMM and how such a model can be implemented into a computational model of tone perception.

3.1 The Model

For every HMM there are three main components: the initial state probability π_i, state transition matrix A, and state emission matrix B. The model μ is represented as a set of these three components in (3.1).

$$\mu = [\pi_i, A, B] \tag{3.1}$$

Both the state of the model and the transitions from one state to the next are considered to be hidden. The state emissions are the visible parts of the model; state emissions are directly linked

to the state transitions. Before I discuss each of these components in more detail, below is a simple example of an HMM called the *Crazy Pop Machine*.

3.1.1 The Crazy Pop Machine Model

The Crazy Pop Machine is a Bayesian model, in the sense that it matches observations to underlying parameters of conditional probabilities. What follows is my version of this standard example (Manning & Schütze, 1999).

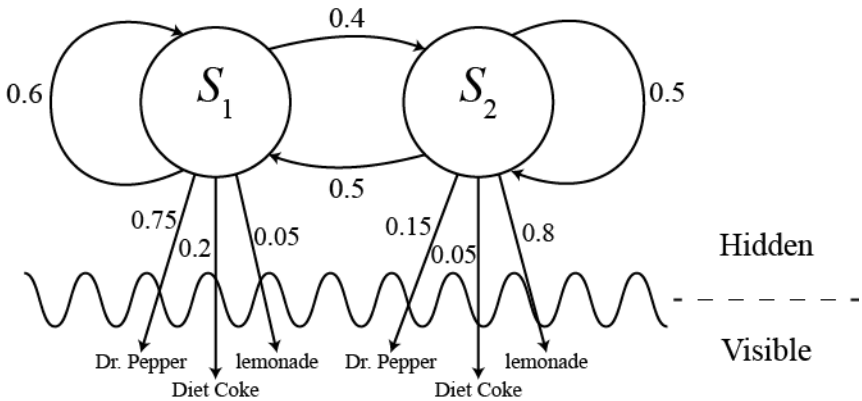

Figure 3.1 The Crazy Pop Machine

Figure 3.1 graphically represents the crazy pop machine model. The crazy pop machine does not work the way a normal, deterministic pop machine would. Instead it spits out beverages with different probabilities, depending on what state it is in.

The crazy pop machine can be in either of two different states of preferences at one time, which are the pop preference or the juice preference. For example, from state S_1, the model transitions to state S_2 40% of the time, or transitions to the same state, S_1, 60% of the time. These transitions must occur with each step in time. It is possible to transition from the pop preference state, S_1, to the juice preference state, S_2, or vice versa.

Either preference represents an underlying hidden conditional parameter for the behavior of the pop machine at a given time. The model only emits a beverage when undergoing a state

transition. When transitioning to the pop preference state, S_1, the machine emits Dr. Pepper 75% of the time, Diet Coke 20% of the time, and Lemonade 5% of the time. These emissions change dramatically in the juice preference state, S_2. When transitioing to state S_2, the machine emits Dr. Pepper 15% of the time, Diet Coke 5% of the time, and Lemonade 80% of the time.

The states and emission probabilities above the wavy line in Figure 3.1 represent the "hidden" part of the model. These are the parameters by which the pop machine determines its output. Only the output is not hidden. A user of the crazy pop machine would never know, at any given time, if the machine were in a pop or a juice preference state. However, they could make a reasonable guess based on the output. This is the Bayesian part of the analysis.

Table 3.1 shows these values for the state transitions from one preference to another. This table gives the probabilities of the state transition for the model. Table 3.2 shows the probabilities of the output of different beverages, based on the preference state of the model.

	Pop Preference S_1	Juice Preference S_2
S_1	0.6	0.4
S_2	0.5	0.5

Table 3.1 Crazy Pop Machine's state transition probabilities

	Dr. Pepper	Diet Coke	lemonade
S_1	0.75	0.2	0.05
S_2	0.15	0.05	0.8

Table 3.2 Crazy Pop Machine output emission probabilites

One more element that is essential to the crazy pop machine is the starting point. This is the initial state matrix that is defined as π_i. The initial state matrix in (3.2) is a matrix with a single row and a number of columns equal to the number of states in the model. To illustrate this, the initial probability of starting in the pop preference state is the matrix value that is in the

first column, which is represented as π_1, and it has an initial state probability of 0.9. The initial probability for starting in the juice preference state is in the second column, which is represented as π_2, which has a probability of 0.1.

$$\pi_i = [0.9 \quad 0.1] \tag{3.2}$$

The probabilities in Table 3.1 and Table 3.2 comprise the other two components of the Crazy Pop Machine model. Matrix (3.3) is the transition matrix A, and (3.4) is the emission matrix B.

$$A = \begin{bmatrix} 0.6 & 0.4 \\ 0.5 & 0.5 \end{bmatrix} \tag{3.3}$$

$$B = \begin{bmatrix} 0.75 & 0.2 & 0.05 \\ 0.15 & 0.05 & 0.8 \end{bmatrix} \tag{3.4}$$

All three of these component matrices make up the model in (3.5).

$$\mu = [\pi_i, A, B] \tag{3.5}$$

With this model and its components, the probability of an observation can be calculated using the Crazy Pop Machine as a HMM. For example, the sequence of symbols in (3.6) represents the beverages that the machine gives at each time interval in the example observation sequence O.

$$O = \{Dr.Pepper, Dr.Pepper, lemonade\} \tag{3.6}$$

The maximum time interval in (3.6) is equal to the total number of observations made, which in this case is $T = 3$. The equation in (3.7) is used to decode the model based on the observation sequence in (3.6). Two probabilities factor into the observation at each time interval in this equation. $a_{X_t X_{t+1}}$ is the probability of transitioning from state X at time interval t to another

state X at the next time interval $t+1$. $b_{X_{t+1}o_t}$ is the probability of the emission of a symbol o_t corresponding to the state transitioned to at the next time interval.

$$P(O|\mu) = \pi_i \sum_{X_1 \ldots X_{T+1}} \prod_{t=1}^{T} a_{X_t X_{t+1}} b_{X_{t+1} o_t} \tag{3.7}$$

This equation is known as *Decoding* (Manning & Schütze, 1999). Its derivation is discussed in section 3.3.1. It is used here to calculate the probability that sequence (3.6) can be generated by the Crazy Pop Machine model.

Decoding requires that every possible sequence of states of the form $X_1 \ldots X_{T+1}$ be tested in the model. Each of these possible state sequences is indexed in the transition matrix in (3.3) as rows, i, and columns, j. Also at each transition there is an emission of a symbol that is found in the sequence of the symbols of O in (3.6). Each emission is indexed in the emission matrix in (3.4) as a row for the state, i, and a column for the emission, j. The probability $P(O|\mu)$ in (3.8) is calculated as the sum of the product of all possible state sequences and emissions from either of the two possible initial states. Logically, decoding just determines the probability of every possible pathway (i.e., every state transition sequence) that could produce the observed sequence of symbols.

$$P(O|\mu) = \pi_i \begin{pmatrix} [a_{11}\, b_{11} \times a_{11} b_{11} \times a_{11} b_{13}] \\ +[a_{11} b_{11} \times a_{11} b_{11} \times a_{12} b_{23}] \\ +[a_{11} b_{11} \times a_{12} b_{21} \times a_{21} b_{13}] \\ +[a_{11} b_{11} \times a_{12} b_{21} \times a_{22} b_{23}] \\ +[a_{12} b_{21} \times a_{22} b_{21} \times a_{22} b_{23}] \\ +[a_{12} b_{21} \times a_{22} b_{21} \times a_{21} b_{13}] \\ +[a_{12} b_{21} \times a_{21} b_{11} \times a_{12} b_{23}] \\ +[a_{12} b_{21} \times a_{21} b_{11} \times a_{11} b_{13}] \end{pmatrix} \tag{3.8}$$

Assuming that the initial state, π_i, starts in S_1 (the pop preference state) we substitute the relevant values for $a_{ij} b_{ij}$ from the transmission matrix A in (3.3) and from the emission matrix

B in (3.4). For example, a_{21} represents a transition from S_2 to S_1, and b_{11} represents the emission of Dr. Pepper from the transition to S_1.

$$P(O|\mu) = \pi_i \begin{pmatrix} [0.6 \times 0.75 \times 0.6 \times 0.75 \times 0.6 \times 0.05] + \\ [0.6 \times 0.75 \times 0.6 \times 0.75 \times 0.4 \times 0.8] + \\ [0.6 \times 0.75 \times 0.4 \times 0.15 \times 0.5 \times 0.05] + \\ [0.6 \times 0.75 \times 0.4 \times 0.15 \times 0.5 \times 0.8] + \\ [0.4 \times 0.15 \times 0.5 \times 0.15 \times 0.5 \times 0.8] + \\ [0.4 \times 0.15 \times 0.5 \times 0.15 \times 0.5 \times 0.05] + \\ [0.4 \times 0.15 \times 0.5 \times 0.75 \times 0.4 \times 0.8] + \\ [0.4 \times 0.15 \times 0.5 \times 0.75 \times 0.6 \times 0.05] \end{pmatrix} \tag{3.9}$$

And then sum the products and multiply by the probability of the initial state being 0.9:

$$\begin{aligned} &= 0.9 \times 0.09214 \\ &= 0.08293 \end{aligned} \tag{3.10}$$

The summation of the products times the initial state sequence gives a probability of 8.29% that the observation sequence (3.6) can be generated by the model.

3.2 The Components

In this section, I discuss the states and symbols of the computational model. Transitions between states compose the transition matrix, and the symbols that are emitted at each state transition compose the emission matrix. The method for how these matrices are computed is called training, and a formal discussion of the training method to calculate the probabilities for each of these matrices is called *Expectation Maximization* (Baum, Petrie, Soules, & Weiss, 1970; Dempster, Laird, & Rubin, 1977). This is discussed in more detail in section 3.3.2.

3.2.1 States

In a HMM, the transition matrix A is a stochastic matrix, which means that for each row the sum of all columns equals 1. A is also a square matrix, as the number of rows is equal to the number of columns. It has the dimensions $m \times m$. There is no pre-determined number of states that a matrix can have or should have, but simply the number of states necessary to adequately

quantify all the possible transitions in the model that are representative of the phenomenon being described (Rabiner, 1989; Manning & Schütze, 1999). Simple HMMs can have as few as 2 to 3 states, and more complex models can have up to 30 states or more.

Table 3.3 is an example of the first two states of a possible transition matrix for the tone perception HMM. This matrix represents the state-to-state transitions for a falling tone. Each state in the matrix represents a different pitch level. In this matrix there are 13 states, including the initial state. Thus, the dimensions of the matrix are 13×13. See section 3.2.3 for discussion about the initial state. It is important to emphasize that this is only a portion of a square matrix, and the probabilities of state transitions shown in Table 3.3 represent only a portion of the complete transition matrix[10].

State/State	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
S_0	0.000	0.000	0.000	0.000	0.000	0.609	0.000	0.000	0.000	0.005	0.377	0.008	0.000
S_1	0.000	0.188	0.062	0.167	0.167	0.104	0.021	0.125	0.062	0.000	0.021	0.042	0.042

Table 3.3 Example Tone Perception transition matrix showing states $S0$ and $S1$

The probabilities are calculated by essentially counting all the different state to state transitions (i.e., pitch transitions) that are observed from the training tokens presented to the model.

[10] The values as presented in Table 3.3 were rounded to 3 decimal places, which was done for ease of presentation. However as presented here, this results in a rounding error of 0.999 and 1.001. This rounding error may also be observed in Table 3.7. In the actual computational model, the values were not rounded to ensure a strict adherence to stochastic constraints in the model.

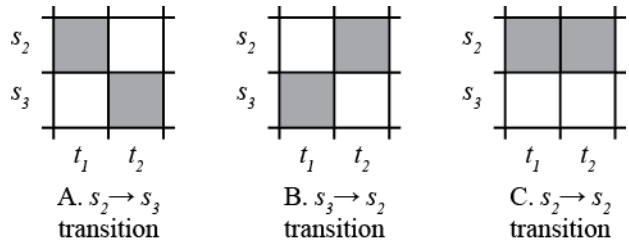

Figure 3.2 State-to-state transitions

Figure 3.2 shows three quarter-sectioned boxes, each representing a possible state-to-state transition between t_1 and t_2. The shaded quarter-sections in each set of boxes represent a transition from one state to another state. Each quarter-section represents a state, s_X, and each box as a whole represents an observed state-to-state transition that occurs between the times $t_1$and t_2 in the sequence. The vertical axis of each box is labeled s_2 and s_3, representing the states.

In Figure 3.2, along the vertical dimension each row represents a quantized unit of frequency. The quantization factor is set as a parameter when the model is created, and can be adjusted for each time the model is run. As I have created the model, the quantization factor is set to $20\ Hz$ per state-to-state transition. I chose this value to limit the number of possible state transitions in the model to a manageable amount for the computational load of the model. In the model, the minimum frequency was set at $80\ Hz$ and the maximum was set at $320\ Hz$. Since the quantization factor was set at $20\ Hz$, every increment in this factor represents a state in the model. For example, S_1 ranged from $80\ Hz$ to $100\ Hz$, S_2 ranged from $100\ Hz$ to $120\ Hz$, etc. The quantization factor is set when the model will run, and it can be changed to produce either more states (e.g., at ranges of $10\ Hz$) or fewer states (e.g., at ranges of $50\ Hz$). Set at $20\ Hz$, this gives the model a total range of $240\ Hz$ for 12 states, plus 1 more state which includes the initial state. When the pitch of a token's lexical tone crosses a quantization boundary, a new

state in the model is recorded. For example a token that has a frequency of $129.84\ Hz$ at t_1 and a frequency of $195.11\ Hz$ at t_2 would be a change of state from S_2 to S_5.

In box A of Figure 3.2, the transition reflects a state change, in this case the time change is t_1 to t_2, and a change from S_2 to S_3 occurs. In box B, the transition is S_3 to S_2, and in C, the transition is S_2 to S_2. The transition matrix includes situations where the time changes but the state does not.

To develop these matrices, the time course of pitch changes of each lexical word in the training database was computed in *Praat* via its autocorrelation[11] pitch tracking function. For each lexical token in the database, pitch was calculated at every $10\ ms$ time interval. These values were used to create the state-to-state transitions, which were stored in a *codebook*. These values thus became a sequence of frequency measurements arrayed in a linear temporal order.

129.8412	195.1070	211.8774	206.8005	209.1749	232.3845	F0 (Hz)
S2	S5	S6	S6	S6	S7	
π_i	t_1	t_2	t_3	t_4	t_5	

Table 3.4 Example Frequency sequence

Table 3.4 is the sequence of pitch values for the high lexical tone for the Thai word /tʰɔːŋ/, 'stomach'. The frequencies are measured in Hertz, and below each value is the specific state for each frequency. The computational model uses sequences like these to form a codebook of transitions. Each transition has a time element that corresponds to each change in state in the sequence, and can be represented in Table 3.5

[11] Autocorrelation is a fairly complex algorithm and its details are not discussed in this dissertation. See Boersma (1993) for a full discussion on the autocorrelation method for calculating periodicity of fundamental frequency in speech perception.

Time	State transitions
t1	S2 → S5
t2	S5 → S6
t3	S6 → S6
t4	S6 → S6
t5	S6 → S7

Table 3.5 State transitions and time

The frequencies in the sequence are in a linear order, but with proportional time intervals for the token rather than incremental (or absolute) time intervals. This coarse-grained representation of time helped reduce the computational load on the model in comparison to one which included frequency data at all absolute time intervals in the token[12]. These proportional intervals include the first time point in the token, time points a tenth of the way through the token[13], one quarter of the way through the token, the midpoint of the token, three quarters through the token, and then at the final time point of the token. There are thus six values in every sequence, which gives five transitions between time points, as shown in Table 3.5. Its calculation is discussed in section 3.2.3. Otherwise, there is no principled reason why representations with more acoustic detail could not be used.

From the transitions in Table 3.4, the model creates a state-to-state transition matrix. Starting with the initial state frequency, π_i, and the second frequency, $f(t_1)$, the model determines what state the frequencies represent, according to the predetermined quantization

[12] Since the acoustic analysis in chapter two showed more change in the latter half of the contour tones, a consideration for future models would be to re-shape the data with more time intervals in the second part of the stimuli.
[13] Using this time point makes the representation asymmetric. It was included as a time point interval so that the initial state and the first transition would not necessarily be the exact same frequency value.

factor. Then the state-to-state transition is recorded in the codebook. The model records all successive transitions in the codebook until the end of the sequence.

3.2.2 Symbols

For every state-to-state transition, a symbol is emitted by the model. These symbols represent the observable part of a model and the probabilities of them being emitted by the model are stored in the emission matrix. The emission matrix obeys stochastic constraints. The number of rows in the matrix equals the same number of states as in the transition matrix, and the number of columns equals the set of symbols. Thus the emission matrix is not necessarily a square matrix. For example, in the tone perception model, the columns equal the emitted symbols which represent the perceived pitch levels that a listener might actually observe, and the rows equal the states of the model.

One common method of transcribing contour tones in Thai and other tone languages is to utilize five equally spaced pitch levels in the normal range of a speaker's voice (Ladefoged & Johnson, 2011). Level 1 represents the lowest pitch and level 5 the highest. For example, the *214* notation associated with the Thai word /na:^{214}/, 'thick', represents that the syllable has a falling-rising pitch contour. This notation represents the rising tone category in Thai. This method of transcribing Thai tones acts as an inspiration for the output symbols used in the emission matrix for the computational model.[14] In order to complete the analogy, the specific acoustic F0 levels for each token first had to be normalized for each speaker's typical F0 range.

For the tone perception model, five speakers were used, and each had varying ranges in pitch as seen in Table 2.4 in Chapter Two. Lobanov's (1971) method for frequency

[14] This is not the only way to do this. For instance, a computer could in principle decode a lot more detail than this.

normalization of vowel formants produced by multiple speakers was adapted for this tone perception model to normalize the F0 frequency of each speaker's lexical tones.

$$F(x) = \frac{x - \mu}{\sigma} \tag{3.11}$$

In (3.11), x is the speaker's current F0 value to be normalized, and μ and σ are the mean and standard deviation of F0 values for the speaker. $F(x)$ can either be a positive or negative number. It is a z-score representing how many standard deviations above or below the mean F0 for that speaker the F0 value is. The value of $F(x)$ is thus a normalized score.

The normalized score is then converted into a tone-level symbol for each lexical tone. These tone symbols were inspired from Chao's (1930) method for transcribing the tonal system of Cantonese, Mandarin and other Asian tone languages. These symbols represent the relative values for each speaker with 1 being the bottom of her F0 range, 3 being the mid of her F0 range, and 5 being the top of her F0 range. Table 3.6 shows the conversion criteria[15].

Symbol	$F(x)$
5	$x > 1.5$
4	$1.5 > x \geq 0.5$
3	$0.5 > x \geq -0.5$
2	$-0.5 > x \geq -1.5$
1	$x < -1.5$

Table 3.6 Converted symbols from $F(x)$ scores

The symbols in Table 3.6 are based on the normalized F0 values. For example, speaker f6 has mean and standard deviation values of $\mu = 166.95\ Hz$ and $\sigma = 20.24\ Hz$. For the falling tone token, *631-phlaa-f6*, the sequence of F0 values are: $206.04\ Hz$, $193.91\ Hz$, $196.95\ Hz$,

[15] However, one problem with this is that if the F0 vowels really are normally distributed, then it would be far more likely to get values in the middle of this range, rather than at the edges. This can be re-parameterized in future versions of the model.

$170.57\ Hz$, $144.14\ Hz$. Each value is inserted into equation (3.11) for $F(x)$ giving the following sequence of z-scores: $1.93, 1.33, 1.48, 0.18, -1.13$. These z-scores are converted, as per Table 3.6, into the following sequence of symbols: $5, 4, 4, 3, 2$.

At each state in the HMM, there is a probability that one of the tone symbols in (3.12) is emitted. Using these symbols, I related raw F0 frequency to normalized tone levels for each speaker.

$$S = \{1, 2, 3, 4, 5\} \tag{3.12}$$

Table 3.7 is an example emission matrix showing the emission probabilities for the first three states for a falling tone. Each row represents a state and each column represents the probability that a symbol is emitted when transitioning into that state. The tone symbols $\{1, \ldots, 5\}$ are represented at the top of each column.

States	1	2	3	4	5
S_1	0.167	0.278	0.056	0.000	0.500
S_2	0.043	0.087	0.174	0.391	0.304

Table 3.7 Example emission matrix and tone symbols

The dimensions of matrix B are $m \times n$, where n is equal to the number of symbols in S and m represents the number of states. The example emission matrix in Table 3.7 (which only includes the first two states) has a total of 13 states. Therefore, the $m \times n$ dimensions for this matrix are 13×5.

3.2.3 Initial state

The model so far includes both a matrix of state-to-state sequences and a matrix of observations (also known as emissions). The missing element is a beginning point. For a HMM, the beginning point is the set of probabilities that the model may begin in any given state. π_i is a

simple matrix that specifies these probabilities. It is a one-dimensional matrix $(1 \times m)$ with length m for the number of states in the HMM.

An example of an initial state matrix is given in Table 3.8. All sequences of state-to-state transitions begin from this state, and no transition can revert back to the initial state. Thus, based on the initial state matrix shown in Table 3.8, the model's initial transition from S_0 has a 65.3% probability of transitioning into S_7, or a 34.4% probability of transitioning into state S_8, and much, much smaller probabilities of transitioning into all of the other states.

State/State	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
S_0	0.000	0.000	0.000	0.000	0.001	0.002	0.653	0.344	0.000	0.000	0.000	0.000

Table 3.8 Example Initial State

3.2.4 Observation sequence

Another important component to a HMM, though not strictly part of the model μ, is an observation sequence O. The observation sequence is a temporally ordered sequence of symbols, as shown in (3.13). These symbols come from the same set of symbols used in the emission matrix.

$$O = \{o_1, o_2, \dots, o_T\} \tag{3.13}$$

The observation sequence is not a component of the HMM, but rather the observations that are presented for analysis to the computational model. In this model, the observation sequence represents the test token with a sequence of symbols. The model computes the likelihood of observing that observation sequence. The sequence O has a time T that equals the number of observations in the sequence.

3.2.5 State sequence

In a HMM, the model passes through a series of states. However, it is unknown at any time what exact state the model is in. Therefore, when we compute the probability of an observation given the model, we have to compute the probability for each state and for every possible state sequence that could produce that observation. A state sequence $X = (X_1, \dots, X_T)$ has a length in transitions which is equal to the length of the observation sequence, O. The initial state is not part of the state sequence, but it is present in the model in order to ensure the number of state transitions correspond to the number of observations.

Within any given model there are N possible states, T observations, and N^T possible state sequences to be calculated. The probability of any given observation is based on the state-to-state transition probability and the state emission probability which may have produced it. The total number of calculations for a given observation sequence with time T and states N in a model is calculated as $(2T - 1)N^T$ (Rabiner, 1989; Manning & Schütze, 1999). For example, if a model has 13 states and 5 observations, then there are $(2 \cdot 5 - 1) \cdot 13^5$ or $3{,}341{,}637$ calculations to be made. Since there is an exponential increase in calculations that is directly related to the length of the observation sequence, the length of the observation sequences was limited in this model in order to make it more practical.

3.3 Objectives for a HMM

For a HMM, there are two objectives which are known as *Decoding* and *Maximizing* (Rabiner, 1989; Manning & Schütze, 1999)[16]. Decoding is used to calculate the probability that

[16] Another common objective for a HMM is to determine the most likely state sequence for an observation. This is known as the *Viterbi Algorithm*. Since this algorithm has not been used in this study, it is not discussed further in this dissertation, but it is pretty easy to see a potential application for this in speech research.

an observation sequence O is produced by the model. For the computational model implemented in this thesis, I created different sub-models for each tone category. The observation O consisted of a tone symbol sequence, and the probability of that tone sequence being produced by each model of a specific tone category was calculated, using the Decoding process. The tone sequence was then categorized according to whichever model produced the highest probability for that tone sequence.

Maximizing involves calculating the model's components that best predict the probability that an observation can be produced by the model. Maximizing is also referred to as *Training* the model, and this process is performed using training data (Manning & Schütze, 1999). The computational model developed in this thesis was trained to predict the correct tone category for a batch of observation sequences.

3.3.1 Decoding

Decoding is a fairly straightforward, if not a brute force, method for determining the probability of an observation given a model. The following mathematical derivation for decoding a model for a given observation follows from Manning & Schütze (1999). The following symbols in (3.14) are used for the decoding derivation:

$$O = An\ observation\ sequence \tag{3.14}$$

$$X = A\ state\ sequence$$

$$\mu = A\ model$$

$$A = A\ transition\ matrix$$

$$B = An\ emission\ matrix$$

$$X_t = A\ state\ in\ the\ state\ sequence\ X$$

$$o_t = A\ temporally\ ordered\ observation$$

$$T = total\ time\ of\ the\ observation\ sequence$$

$$t = an\ instance\ in\ the\ observation\ sequence$$

$$b_{X_{t+1}o_t} = An\ emitted\ symbol\ from\ state\ X_t\ to\ X_{t+1}\ and\ o_t$$

$$a_{X_tX_{t+1}} = A\ transition\ from\ state\ X_t\ to\ X_{t+1}$$

$$\pi_i = Initial\ state\ probability$$

If X represents any possible temporal state sequence where $X = (X_1, \dots, X_{T+1})$, then the probability of an observation o_t given a sequence X_t, X_{t+1} and a model μ is calculated as:

$$\begin{aligned} P(O|X,\mu) &= P(o_t|X_t, X_{t+1}, \mu) \\ &= b_{X_2o_1} b_{X_3o_2} \dots b_{X_{T+1}o_T} \end{aligned} \tag{3.15}$$

This is calculated for the whole observation sequence. For $P(O|X,\mu)$, it is the product of all the emissions of symbols b for every $x_{t+1}o_t$ in state sequence X. The emission symbol o_t and the probability b are relative to the next state X_{t+1} in the sequence. The probability of a state sequence X, given a model, is the initial state probability times the product of the probability of transitions for every sequence x_tx_{t+1} in the state sequence X.

$$P(X|\mu) = \pi_{X_0} a_{X_1X_2} a_{X_2X_3} \cdots a_{X_TX_{T+1}} \tag{3.16}$$

The probability of an observation sequence and a state sequence given a model is the product of the equations in (3.15) and (3.16).

$$P(O, X|\mu) = P(O|X, \mu)P(X|\mu) \tag{3.17}$$

Therefore, to get the probability of the observation sequence given the model, equation (3.18) is the sum of all possible state sequences X in equation (3.17). This is the sum of all the products of probabilities for every possible state sequence that could lead to that observation sequence.

$$\begin{aligned} P(O|\mu) &= \sum_{X} P(O|X, \mu)P(X|\mu) \\ &= \pi_i \sum_{X_1 \ldots X_{T+1}} \prod_{t=1}^{T} a_{X_t X_{t+1}} b_{X_{t+1} o_t} \end{aligned} \tag{3.18}$$

3.3.2 Expectation Maximization algorithm

The Baum-Welch algorithm (Baum, Petrie, Soules, & Weiss, 1970; Rabiner, 1989), also known as the Expectation Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977), is one method used to train a HMM. The purpose of this algorithm is to create a model with the best probability of emitting an observation sequence by a given HMM. Each component of a model ($\mu = [\pi_i, A, B]$) can be modified to reflect a trained model, $\overline{\mu} = [\bar{\pi}_i, \bar{A}, \bar{B}]$.

For the tone perception model, maximized values for each component can be computed. For example, maximization of the initial state probability, π_i, occurs by taking the initial frequency, when $t = 1$, of each training token and summing the number of times that the initial frequency equals each state, S_i. N represents the total number of states for the model, which (in this case) equals 13. When an initial frequency corresponds to a specific state, S_i, it is recorded in $\gamma_1(i)$.

$$\gamma_1(i) = Total\ number\ of\ observations\ in\ state\ S_i\ at\ time\ (t = 1) \qquad (3.19)$$

The trained initial state probability $\bar{\pi}_i$ in (3.20) is the set of the total number of observations made at each state S_i, divided by the sum of all observations made, $\sum_{i=1}^{N} \gamma_1(i)$.

$$\bar{\pi}_i = \frac{\gamma_1(i)}{\sum_{i=1}^{N} \gamma_1(i)} \qquad (3.20)$$

Maximization of the transition matrix $\bar{A}$ is done in a similar manner as the maximization of the initial state $\bar{\pi}_i$, using the same data. However, instead of just the initial frequency, observations are counted for all state-to-state transition sequences from S_i to S_j for all times t for each training token. The total number of observed transitions for a state-to-state transition is recorded as $\xi_t(i,j)$, which is informally defined in (3.21).

$$\xi_t(i,j) = Total\ number\ of\ observed\ transitions\ from\ S_i\ to\ S_j\ for\ each\ t \qquad (3.21)$$

For calculating the maximized state-to-state transition $\bar{a}_{ij}$ in (3.22), the numerator is the total number of observations made from S_i to S_j at each time t in the transition sequence S, which equals $\xi_t(i,j)$. The denominator is similar to the initial probability given in (3.20), except that the total number of observations for a given state must also be summed over all possible time points. The sums go to $T - 1$ because there is no transition after the last time point, T. $\gamma_t(i)$ is the total number of times in state i, regardless of any transitions to state j. This gives the probability for each state-to-state transition in the transition matrix $\bar{A}$.

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{i=1}^{N} \gamma_t(i)} \qquad (3.22)$$

The emission matrix $\bar{B}$ is trained in a similar fashion as the state-to-state transition matrix $\bar{A}$, with a major difference being that the symbols are counted instead of the state-to-state

transitions. The total number of observed symbols for each time t in the state sequence is recorded as $\delta_t(i,k)$ and defined in (3.23).

$$\delta_t(i,k) = Total\ number\ of\ observations\ of\ symbol\ k\ in\ S_i\ for\ each\ time\ t \qquad (3.23)$$

$$\bar{b}_{ik} = \frac{\sum_{t=1}^{T} \delta_t(i,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(i)} \qquad (3.24)$$

The numerator in (3.24) is the sum for all T of observed symbols k that are in state S_i. The denominator is nearly identical to (3.22) except that M is equal to the total number of symbols instead of states, and $\gamma_t(i)$ is the total number of observations of symbols in state i regardless of symbol k.

A description of how a model using these algorithms for tone identification was trained and tested is presented in chapter Four.

Chapter Four: Computational Model

In this chapter I discuss the framework in which I implemented the *Expectation Maximization* and *Decoding* algorithms that were discussed in chapter Three. With this framework I tested how well the model identified the tone category of the Thai test tokens. I tested two factors: token type and quantization factor. I also tested the model's ability to categorize each token's tone category by using the simplified and multiple speaker databases. In total I ran four separate experiments using the framework presented in this chapter. The first two experiments tested the model's performance on the single speaker database. These two experiments differed based on quantization factor: experiment one used a quantization factor of $20\ Hz$, and experiment two used a quantization factor of $10\ Hz$. The last two experiments tested the model's performance on the multiple speaker database, with experiment three using only citation tokens and experiment four using both citation and continuous tokens. Both of these experiments used a quantization factor of $20\ Hz$.

4.1 Computational Model

The framework shown in Figure 4.1 represents the paradigm for testing the model. This paradigm was inspired by earlier perceptual training experiments, including Wayland and Guion (2003) and Wayland and Li (2008). The experiments in these studies used a Pre-Test, Training and Post-Test methodology for tone identification.

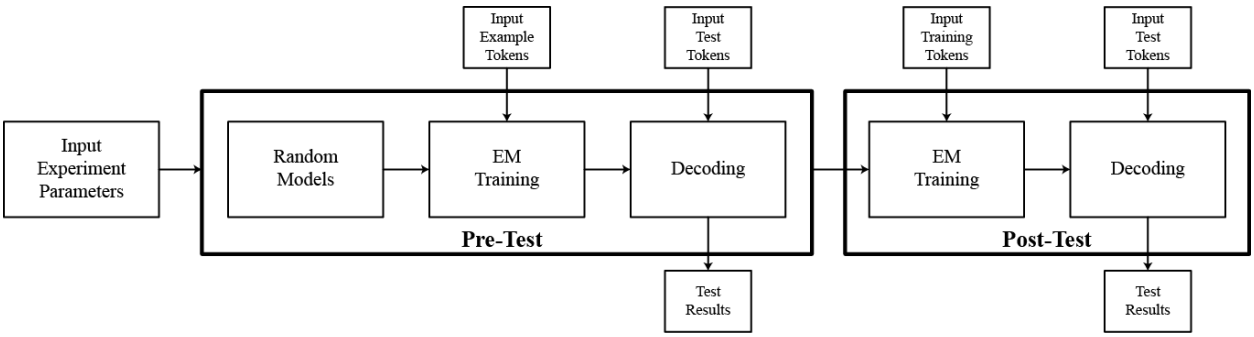

Figure 4.1 Experimental Paradigm

First, I tested an untrained model (in the Pre-Test), then I trained the model, and finally I tested the model again (in the Post-Test) to see how training had improved the model's performance. Each execution of this paradigm is called a simulation, since the paradigm is intended to simulate a human subject's performance in a perceptual training experiment.

The experimental paradigm begins by inputting the experiment's parameters, and is then followed by the Pre-Test and Post-Test phases. The experimental parameters include the quantization parameter and the number of tokens used for training and testing operations.

In the Pre-Test phase, there are three stages, with two input operations and one output operation. *Make Random Models* is the first stage of the Pre-Test phase. It creates five random sub-models, one for each of the lexical tone categories. Each random sub-model includes the three HMM elements, including a transition matrix, an emission matrix and an initial state probability. A random probability was generated for each cell in these three components, but still obeying stochastic constraints for each matrix. These random model components form the basis (or starting point) for each simulation. These elements are matrices that are divided into cells that are indexed via rows and columns, as discussed in chapter three. Each cell in the transition matrix, the emission matrix and in the initial state probability is stored as an individual

variable to be accessed later in the computational model. Next the *Input Example Tokens* operation occurs, where five example tokens, one for each lexical tone category, are used as input into the *Expectation-Maximization (EM) Training* stage of the Pre-Test. The EM data from the example tokens is added to the initial random models (which are not based on the example tokens). This set of example tokens is a pre-determined set of tokens that was the same for all the different simulations[17]. EM Training in the Pre-Test uses these example tokens to train the five random sub-models. The *Input Test Tokens* operation randomly selects an equal number of pre-determined tokens from each of the five lexical tone categories to be tested in the *Decoding* component of the Pre-Test phase. The number of test tokens is set as an initial parameter when running the paradigm. This component uses the HMM Decoding algorithm discussed in chapter three. Each sub-model calculates a probability that a token's tone sequence is predicted by it. In the *Test Results* operation, the sub-model that scores the highest probability for a test token represents the predicted tone category for that token. If the predicted tone category is the same as the actual tone category for that token, then the model has correctly identified the tone category of the test token. This is considered as a hit in the statistical evaluation of the model's performance.

Following the Pre-Test phase, the computational model enters the Post-Test phase. This phase has two stages, with two input operations and one output operation. *Input Training Tokens* is the first operation. It inputs a new set of randomly chosen training tokens into the Post-Test phase's *EM Training* component, to train the five sub-models from the Pre-Test phase. The number of these training tokens is set as a parameter at the beginning of the model, which was

[17] It might also have been a good idea to randomize the set of example tokens as well.

normally set at 30 training tokens from each tone category[18]. Since the set of randomly chosen training tokens is selected from the same database as the test tokens, it is possible for overlap between the training and testing stimuli to occur. At the completion of training, the models are again tested in the *Decoding* component using the same decoding algorithm as the Pre-Test phase. Just as it did during Pre-Test phase, the *Input Test Tokens Operation* randomly selects a new set of random test tokens, which was normally set at 25 tokens from each tone category. The *Test Results Operation* in the Post-Test operates in the same fashion as the Pre-Test operation.

4.2 Experiments

Using the computational testing paradigm in Figure 4.1, four separate experiments were conducted to investigate the effects of different structures of model components on performance, as well as different sets of training and testing stimuli on tone identification performance. Experiments One and Two differed in model components, which resulted in two different sizes for the transition matrices used for each experiment's computational model. Experiments Three and Four differed in the type of stimuli used for training and testing each experiment's model. All experiments were conducted with 50 simulations each.

4.2.1 Analytic Methods Used

Tone identification performance was measured in each experiment with percent correct, D-prime and bias scores. Percent correct was calculated as the number of correct responses over the total number of stimuli presented to each simulation. D-prime and bias scores were calculated for each simulation on a per category basis. Sensitivity and bias measures are

---

[18] More tokens could have been used for training, but I wanted the training tokens used for the model and the training stimuli used for the human perception experiment to be the same number.

borrowed from Signal Detection Theory (Wickens, 2002). D-prime measures sensitivity, which is the actual perceptual distance between response categories. It is calculated by effectively removing response bias from the equation. Response bias is a measure of how likely the model is to select a particular response category over another.

D-prime was calculated as the difference between the percentage of hits and the percentage of false positives in each simulations responses, expressed in terms of z-scores. A "false positive" was recorded any time a response category was incorrectly selected as a response to a tone of a different category. For example if a simulation falsely identified a mid tone token as a high tone category then the incorrect identification would be recorded as a false positive for the high category. Equation (4.1) is the formula for calculating D-prime scores that was used in this dissertation.

$$d' = z(Hits) - z(False\ Positives) \tag{4.1}$$

The D-prime score is calculated for each tone category independently.

Response bias is a similar calculation to D-prime, but only in the sense that it uses normalized z-scores for its calculation. Equation (4.2) is the formula for calculating response bias that was used in this dissertation.

$$Bias = -\frac{1}{2} \times \big(z(H) + z(FP)\big) \tag{4.2}$$

It was calculated as the sum of the normalized portion of hits and false positives times negative one half. A positive bias score represents a bias against responding with that category, and a negative score represents a bias in favor of responding with that category. Essentially, the response bias is independent of what the listener (or the model) actually hears (or perceives) in the signal.

4.2.2 Fine-Grained vs. Coarse-Grained Models

The first two experiments were run to test the effects of the granularity of fundamental frequency representations on the performance of the model. This was done by setting the quantization factor to a smaller frequency resulting in a larger transition matrix for the computational model. *Experiment One* had its quantization factor set at $20\ Hz$. Due to this setting, the computational model in Experiment One was considered to have a *coarse-grained* distinction of fundamental frequency. *Experiment Two*, however, had its quantization factor set at $10\ Hz$. This gave the computational model a more fine-grained distinction in F0 data, so this quantization factor resulted in a larger transition matrix with a total of 26 states. The hypothesis tested in these two experiments was whether more information (in the form of a larger transition matrix) would cause the computational model to perform better on the tone identification task. Thus, in Experiment Two, if the model is more responsive to changes in F0 due to a difference in quantization parameter, then it will perform better on the contour tones (Falling and Rising), since contour tones, as opposed to level tones, exhibit (greater) change in F0.

Experiments One and Two were run using test and training stimuli only from the single speaker database described in chapter two. For both of these experiments, 30 tokens from each tone category were used for training the sub-models in the training phase, and 25 tokens from each tone category were used to test the model in the pre-test and post-test phases.

4.2.2.1 Percent Correct

For each of the experiments, I ran Paired-samples *t*-tests, which showed that the Post-Test mean percent correct scores were significantly greater than the Pre-Test mean scores. Table 4.1 shows the mean percent correct scores for the Pre-Test and Post-Test, and *t*-statistics and *p*-

values for each experiment. Each experiment's training algorithm caused the model to perform significantly better.

Experiment	Pre-test	Post-test	$t(49) =$	p
1	52.7%	78.5%	23.23	< 0.001
2	55.9%	77.6%	22.18	< 0.001

Table 4.1 Mean Performance per Test

4.2.2.2 Comparing Percent Correct Scores between Experiment One and Two

For Experiment One and Two, the difference in quantization factor was tested with a two-way repeated measures ANOVA to test for any main effects on percent correct scores. Test was treated as a within-subjects factor, with Pre-Test and Post-Test as levels. Experiment was treated as a between-subjects factor, with both experiments as factor levels. There was no significant effect of Experiment on percent correct scores, $F(1,98) = 1.99$, $p = 0.161$, but there was a significant effect of Test, $F(1,98) = 1030.01$, $p < 0.001$. There was a significant interaction of Experiment and Test on percent correct scores, $F(1,98) = 22.12$, $p < 0.001$.

Table 4.2 shows the comparison of scores between Experiments and Tests using paired-samples *t*-tests.

Test	Experiment 1	Experiment 2	$t(49) =$	p
Pre-test	52.7%	55.9%	-2.90	= 0.006
Post-test	78.5%	77.6%	0.78	= 0.439

Table 4.2 Mean performance for Experiment One and Two

The mean percent correct score for the Pre-test in Experiment One was significantly lower than the mean percent correct score for the Pre-Test in Experiment Two. However, in the Post-test the mean percent correct score for Experiment One was not significantly different from the mean percent correct score for Experiment Two. Quantization may have had a significant effect on the Pre-Tests for both experiments. A smaller quantization factor helped the model perform better in

the Pre-Test for Experiment Two. Its computational model had a larger transition matrix, which enabled the model to make more *fine-grained* distinctions during the test, allowing it to deal with variability better. Thus, a larger transition matrix may cause the model to be more sensitive to distinctions between test stimuli prior to EM training. However, after EM training the benefits of a larger transition matrix no longer had a significant effect on performance scores. Additional information from training stimuli added more variability into the model during EM training, causing no significant difference in performance scores during the Post-Test. Perhaps, more information and more variability made it harder to generalize patterns.

Although, another interpretation may be that the two Post-tests were significantly better than the two Pre-Tests, so it was not harder to generalize patterns. It was just that the EM training was so effective that the benefit of having a more fine-grained model was overridden. Perhaps the advantages of the finer-grained distinctions would likely emerge with larger testing and training sets.

4.2.2.3 Experiment One's response to specific tone categories

During Experiment One, each simulation was presented with 25 tokens for each of the 5 lexical tone categories. Table 4.3 is a confusion matrix which shows the Pre-Test results, and Table 4.4 shows the Post-Test results from Experiment One. The rows represent the responses to each stimulus type, and the columns represent the number of times each response type was given. The Percent Correct column gives the percentage of correctly identified tokens per category for all simulations in the experiment. Table 4.4 also includes a column showing the difference in Percent Correct scores between the Pre-Test and the Post-Test.

	H	M	L	R	F	Total	Percent Correct
H	196	740	0	16	298	1250	15.7%
M	216	985	10	3	36	1250	78.8%
L	10	47	1137	39	17	1250	91.0%
R	1	3	1058	131	57	1250	10.5%
F	112	34	50	212	842	1250	67.4%
Total	535	1809	2255	401	1250	6250	

Table 4.3 Experiment one Pre-Test

	H	M	L	R	F	Total	Percent Correct	Change
H	853	245	0	2	150	1250	68.2%	52.5%
M	200	969	59	22	0	1250	77.5%	-1.3%
L	0	3	1096	151	0	1250	87.7%	-3.3%
R	2	1	103	1129	15	1250	90.3%	79.8%
F	345	8	6	34	857	1250	68.6%	1.2%
Total	1400	1226	1264	1338	1022	6250		

Table 4.4 Experiment one Post-Test

For Experiment One, I ran two two-way repeated-measures ANOVAs to test the effects of Test Type and Tone Category on D-prime and response bias. Category and Test were the within-subjects factors in the analysis. Category had five levels, corresponding to each of the five lexical tone categories, and test had two levels, corresponding to the Pre-Test and the Post-Test. There was a significant effect of Category on D-prime, $F(4,196) = 78.37$, $p < 0.001$, and there was also a significant effect of Test on D-prime, $F(1,49) = 242.20$, $p < 0.001$. There was a significant interaction of Category and Test on D-prime, $F(4,196) = 41.63$, $p < 0.001$. There was a significant effect of Category on response bias, $F(4,196) = 44.91$, $p < 0.001$, and there was also a significant effect of Test on response bias, $F(1,49) = 17.27$, $p < 0.001$. There was a significant interaction of Category and Test on response bias, $F(4,196) = 61.52$, $p < 0.001$.

Table 4.5 shows the mean D-prime and response bias scores for each category in the Pre-Test and Post-Test. A paired-samples *t*-test was also performed for each category between test types, and the *t*-statistic and *p*-value for each test is also reported in Table 4.5.

	D-prime				Bias			
Lexical Tones	Pre-Test	Post-Test	$t(49) =$	p	Pre-Test	Post-Test	$t(49) =$	p
High	0.482	1.912	-9.43	< 0.001	1.448	0.423	11.25	< 0.001
Mid	2.330	2.918	-2.87	= 0.006	-0.176	0.569	-3.81	< 0.001
Low	2.551	3.682	-5.79	< 0.001	-0.513	0.182	-6.35	< 0.001
Rising	0.332	3.674	-15.10	< 0.001	1.620	0.165	14.32	< 0.001
Falling	1.906	2.636	-7.40	< 0.001	0.473	0.733	-3.46	= 0.001

Table 4.5 D-prime and Bias scores for Experiment One

The mean D-prime measures for each tone category in the Pre-Test was significantly lower than the mean D-prime measures in the Post-Test. For Experiment One, the model was more sensitive to distinctions in the test stimuli in the Post-Test than it was in the Pre-Test.

The mean response bias for the Mid, Low and Falling tone categories was significantly lower in the Pre-Test than in the Post-Test. In the Post-Test the model was more biased towards making a response for these tone categories. However, the mean response bias for the High and Rising categories was significantly higher in the Pre-Test than in the Post-Test. Thus, in the Post-Test, the model was more inclined to respond with these tone categories.

4.2.2.4 Experiment Two's response to specific tone categories

Table 4.6 and Table 4.7 show the pre-test and post-test confusion matrices for experiment two. Table 4.7 also includes a column showing the difference in Percent Correct scores from the Pre-test to the Post-test.

	H	M	L	R	F	Total	Percent Correct
H	1061	85	3	0	101	1250	84.9%
M	632	578	40	0	0	1250	46.2%
L	21	0	1224	5	0	1250	97.9%
R	76	8	1048	118	0	1250	9.4%
F	440	147	125	28	510	1250	40.8%
Total	2230	818	2440	151	611	6250	

Table 4.6 Experiment two Pre-Test

	H	M	L	R	F	Total	Percent Correct	Change
H	744	415	0	0	91	1250	59.5%	-25.4%
M	64	1075	111	0	0	1250	86.0%	39.8%
L	0	0	1138	112	0	1250	91.0%	-6.9%
R	0	0	184	1065	1	1250	85.2%	75.8%
F	390	0	2	33	825	1250	66.0%	25.2%
Total	1198	1490	1435	1210	917	6250		

Table 4.7 Experiment two Post-Test

Two 2-way ANOVAs were run on all simulations for Experiment Two to test the effects of Test Type and Tone Category on the D-prime and response bias measures. This ANOVA had the same test factors as those run for Experiment One. There was a significant main effect of Category on D-prime measures, $F(4,196) = 63.67$, $p < 0.001$, and there was a significant main effect of Test on D-prime, $F(1,49) = 131.90$, $p < 0.001$. There was a significant interaction of Test and Category on D-prime, $F(4,196) = 44.07$, $p < 0.001$. There was a significant main effect of Category on response bias measures, $F(4,196) = 190.80$, $p < 0.001$, and there was a significant main effect of Test on response bias, $F(1,49) = 146.80$, $p < 0.001$. There was a significant interaction of Test and Category on response bias, $F(4,196) = 112.20$, $p < 0.001$

	D-prime				Bias			
Lexical Tones	Pre-Test	Post-Test	*t*(49) =	*p*	Pre-Test	Post-Test	*t*(49) =	*P*
High[19]	1.980	1.722	2.25	= 0.029	-0.241	0.598	-8.28	< 0.001
Mid	1.790	2.711	-6.29	< 0.001	0.978	0.102	9.47	< 0.001
Low	3.296	3.428	-0.80	= 0.430	-0.948	-0.078	-9.11	< 0.001
Rising	1.250	3.523	-12.86	< 0.001	2.142	0.344	17.72	< 0.001
Falling	1.891	2.842	-7.39	< 0.001	1.292	0.933	3.45	= 0.001

Table 4.8 Performance, D-prime and Bias scores within categories Experiment Two

Table 4.8 shows the mean D-prime and response bias measures for each category and test for Experiment Two. A paired-samples *t*-test was used to compare the measures for each

[19] It is important to point out that the change in D-prime scores for the High category does not reach significance because of the *post-hoc* Bonferonni correction used here. Here, significance is 0.001.

category and test; the *t*-statistics and *p*-values are also reported in Table 4.8. The Mid, Rising and Falling tone categories D-prime measures were significantly lower in the Pre-Test than in the Post-Test. Unlike the model in Experiment One, setting the quantization factor to 10 Hz caused an increase in the model's sensitivity only for these three tone categories in the Post-Test. The biggest increase in D-prime scores was for the Rising category, which is a contour category. The other contour tone category, Falling, also shows a significant increase in D-prime scores. However, the Mid tone category is a level tone category, and for it the model also increased in its sensitivity in favor of this category. High and Low are both level tones, so maybe the *finer-grained* model does not have more success with the contour tones so much as it has more difficulty with the level tones. Although what reason this may be is still unclear to me.

Looking at response bias shows some other interesting patterns, and perhaps offers another piece to the puzzle. In the Post-Test, the model had a significantly greater response bias against the High and Low categories, and the response bias was significantly less for the Mid, Rising and Falling categories.

This pattern of response bias mimics the pattern for the D-prime scores. A pairwise *t*-test using a Bonferroni correction showed that the Pre-Test's mean D-prime score for the Low category was significantly higher than all the other categories. The model was already highly sensitive to the Low category in the Pre-Test, and there was no significant change in sensitivity in the Post-Test. This was also true, although to a lesser degree, for the High category[20].

[20] Although it is different for the High category in that it's on the lower end of sensitivity in the Pre-Test and then doesn't improve in the Post-Test.

In the Pre-Test the model's response bias was, likewise, significantly biased most in favor of the Low and High categories, and in the Post-Test the response bias increased significantly against these categories, but for all other categories the model's response bias significantly changed to be more in favor of these other categories. This may be evidence that with a larger, or more *fine-grained* transition matrix, the model is more sensitive to changes in fundamental frequency, because there is more information about it, and this becomes more evident in the contour tones, particularly in the Rising tone category. It may be that it was easy for the model to clump everything into the High and Low categories at the beginning during the Pre-Test, and then it had more difficulty finding the boundary line between the three level tone categories in training[21].

4.2.2.5 Comparing the model's sensitivity to changes in F0 between experiments

To directly test the evidence that suggests a more fine-grained model is more sensitive to changes in fundamental frequency, I ran a three-way repeated-measures ANOVA comparing the results from these two experiments. In this ANOVA I tested the effects of Experiment, Test Type and Tone Category on D-prime measures for all simulations. Tone and Test were the within-subjects factors, and Experiment was the between-subjects factor. Category and Test used the same factor levels as in the previous analyses, and Experiment had two levels which were Experiment One and Experiment Two. There was a significant main effect of Category on D-prime measures, $F(4,392) = 131.50$, $p < 0.001$, and there was a significant interaction of Experiment and Category, $F(4,392) = 12.70$, $p < 0.001$. There was a significant main effect of

[21] An alternate possibility may be that the model could not sort through the middle of the range when it has too much information to begin with.

Test, $F(1,98) = 374.00$, $p < 0.001$, and there was a significant interaction of Experiment and Test, $F(1,98) = 30.30$, $p < 0.001$. There was a significant interaction of Category and Test, $F(4,392) = 70.16$, $p < 0.001$. There was a significant three-way interaction of Category, Test and Experiment, $F(4,392) = 15.15$, $p < 0.001$.

For the Pre-Test, the mean D-prime measure for Experiment One was significantly lower than for Experiment Two. However, in the post-test there was no significant difference in D-prime scores between experiments. Table 4.9 shows the mean D-prime score and *t*-test statistics of the comparison across both Test types and experiments.

	Pre-Test				Post-Test			
	Experiment 1	Experiment 2	$t(249) =$	p	Experiment 1	Experiment 2	$t(249) =$	p
D-prime	1.520	2.041	-5.88	< 0.001	2.964	2.845	1.63	= 0.105

Table 4.9 D-prime between Tests and Experiments

Looking within categories shows more significant differences between both experiments, and these results are shown in Table 4.10.

D-prime								
	Pre-Test				Post-Test			
Categories	Experiment 1	Experiment 2	$t(49) =$	p	Experiment 1	Experiment 2	$t(49) =$	p
High	0.482	1.980	-10.39	< 0.001	1.912	1.722	1.78	= 0.081
Middle	2.330	1.790	2.29	= 0.026	2.918	2.711	1.29	= 0.203
Low	2.551	3.296	-4.59	< 0.001	3.682	3.428	1.33	= 0.191
Rising	0.332	1.250	-5.06	< 0.001	3.674	3.523	0.66	= 0.510
Falling	1.906	1.891	0.14	= 0.891	2.636	2.842	-2.33	= 0.024

Table 4.10 Tone category D-prime measures between Experiments One and Two

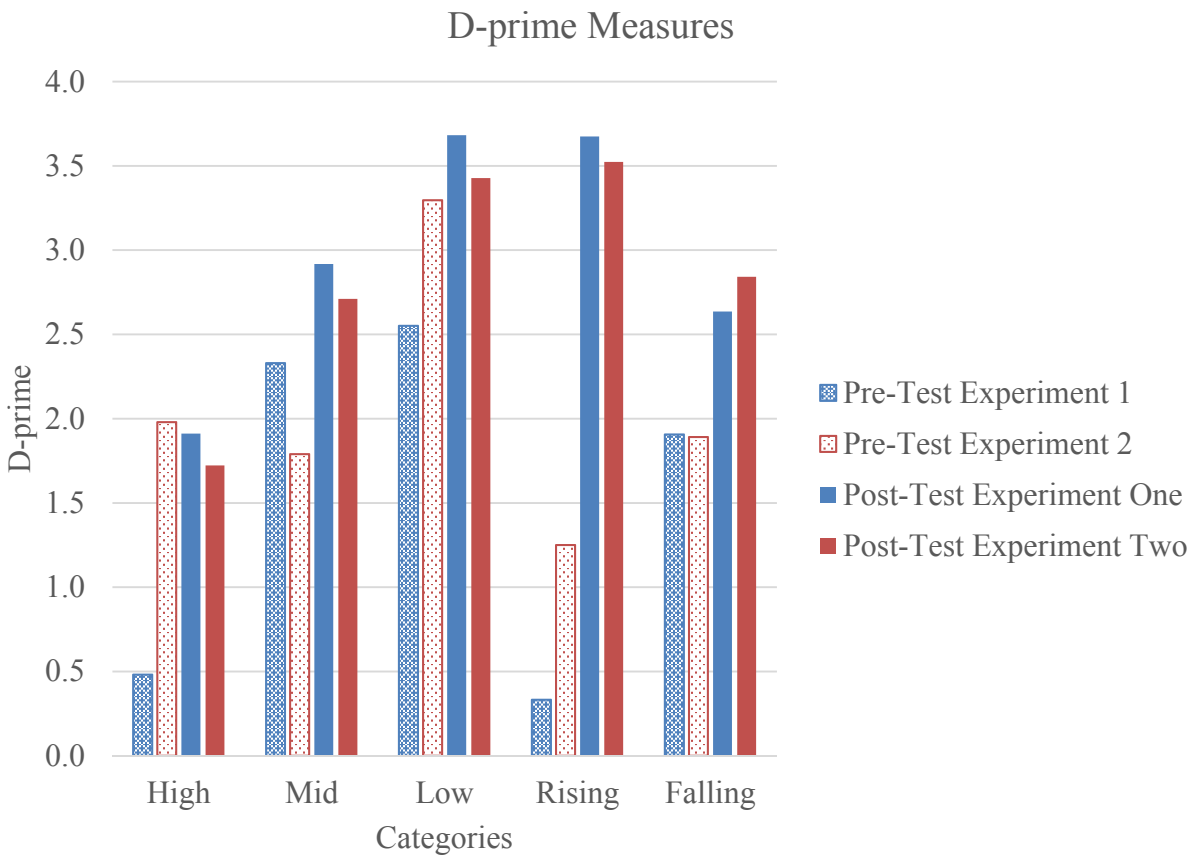

Figure 4.2 Mean D-prime measures for tests between experiments

As shown in Table 4.10, in the Pre-Test there were significant differences in D-prime measures between experiments for all categories except in the falling category. However, in the post-test there were no significant differences in D-prime measures, except for the falling category, whose mean D-prime was significantly lower in Experiment One than in Experiment Two. The Falling category has a falling contour, and the larger matrix size could be the factor that made Experiment Two's model more sensitive to this category[22].

[22] It is important to note that Experiment One caught up to Experiment Two in the Post-Test for the Rising category.

4.2.2.6 Comparing the model bias between Experiments One and Two

I also ran a three-way repeated-measures ANOVA to test the effects of Experiment, Test Type and Tone Category on response bias. This ANOVA used the same test factors and levels as the one in section 4.2.2.5. There was a significant main effect of Category, $F(4,392) = 153.72$, $p < 0.001$, and there was a significant interaction of Experiment and Category, $F(4,392) = 39.02$, $p < 0.001$. There was a significant main effect of Test, $F(1,98) = 93.94$, $p < 0.001$, and there was a significant interaction of Experiment and Test, $F(1,98) = 6.31$, $p = 0.014$. There was a significant interaction of Category and Test, $F(4,392) = 106.96$, $p < 0.001$, and there was a significant three-way interaction of Category, Test and Experiment, $F(4,392) = 57.54$, $p < 0.001$.

Bias

	Pre-Test				Post-Test			
Categories	Experiment 1	Experiment 2	$t(49) =$	p	Experiment 1	Experiment 2	$t(49) =$	p
High	1.448	-0.241	17.82	< 0.001	0.423	0.598	-1.72	= 0.093
Mid	-0.176	0.978	-9.72	< 0.001	0.569	0.102	2.27	= 0.028
Low	-0.513	-0.948	5.19	< 0.001	0.182	-0.078	2.38	= 0.021
Rising	1.620	2.142	-5.95	< 0.001	0.165	0.344	-1.67	= 0.101
Falling	0.473	1.292	-10.22	< 0.001	0.733	0.933	-1.87	= 0.068

Table 4.11 Tone category response bias measures between Experiments

In the Pre-Test, the response bias measures for all categories were significantly different between experiments as seen in Table 4.11. For Experiment Two in the Pre-Test, the model was significantly less biased against the level tone categories for High and Low than the model was in Experiment One. However, for Experiment Two, the model was significantly more biased against the contour tone categories, Falling and Rising. Perhaps the model is not capturing the contours as well, due to too much *fine-grained* information. Surprisingly, however, in Experiment Two the model was also significantly more biased against the Mid tone category than it was in Experiment One. Finally, only the post-test bias for the Mid and Low categories were significantly greater in Experiment One than in Experiment Two.

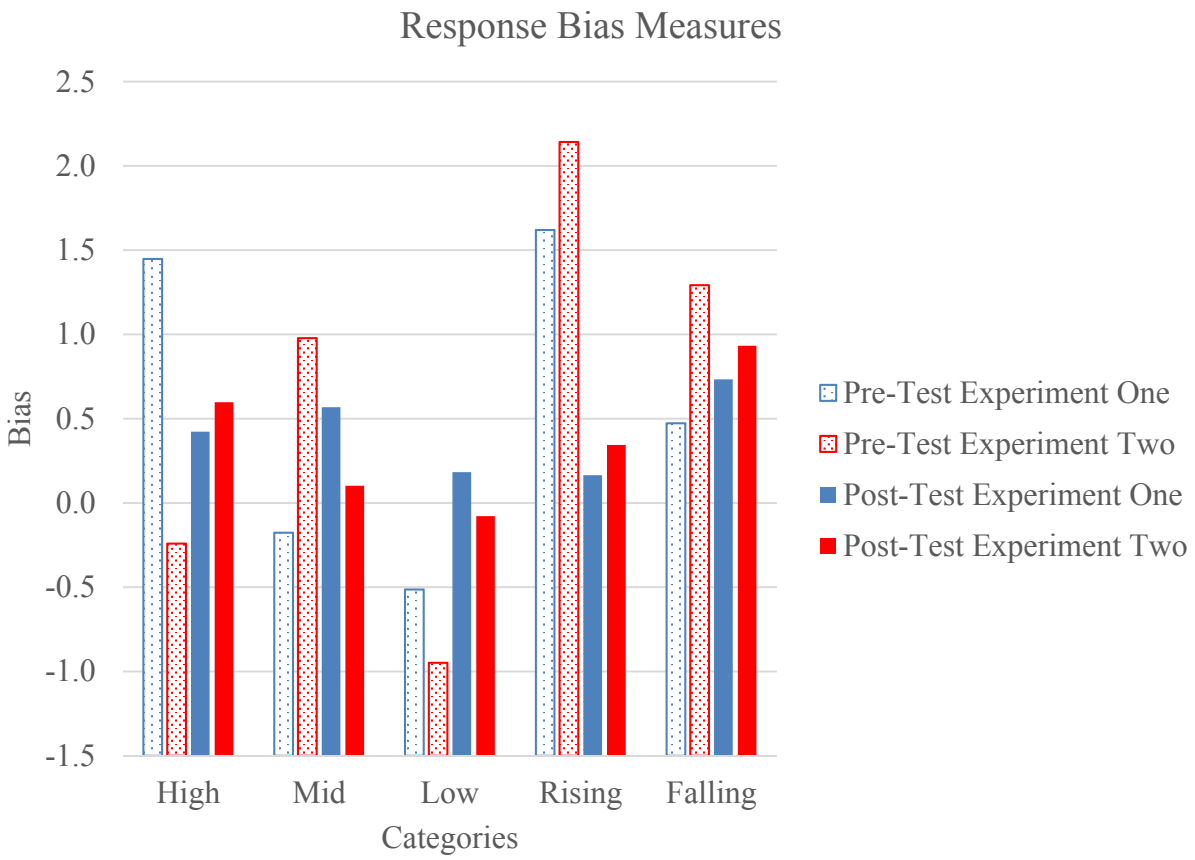

Figure 4.3 Mean response bias measures between experiments

In terms of response bias, the larger, more fine-grained transition matrix provided the model with a greater amount of information about the fundamental frequency. This was shown to be an advantage in Experiment Two for the model in the Pre-Test, particularly with respect to the Rising contour tone category. For the Pre-Test in Experiment Two, the model was sensitive to the Rising tone category, and it was also significantly more biased against this category than in Experiment One. Although, for some reason, in both experiments there was a large degree of bias against this category. Due to a larger transition matrix in Experiment Two's Pre-Test, the model had more correct hits and fewer false positives. In other words, it was more correctly identifying the Rising tone category correctly, as the larger transition matrix allowed the model

to be more sensitive to changes in fundamental frequency. However, in the Post-Test there was no significant advantage with the contour tone categories for Experiment Two, but there was some advantage with the Mid and Low level tone categories. The model was less biased against these categories.

4.2.3 Continuous vs Citation Stimuli Type

The next two experiments were run to test if the context factor had a significant effect on the performance of the computational model in the tone identification task. In Chapter Two, I showed that the Rising and Falling tones had a contour F0 trajectory. The High tone category also had a slight rising contour F0 as well. Furthermore, Thai words with contour tones (Falling and Rising) spoken in a continuous context exhibited a smaller range of change in fundamental frequency over time. This was also shown in the sequential portion graphs of the citation and continuous data in chapter Two. Potisuk, Gandour and Harper (1997) argued that a smaller range in the changes of fundamental frequency make naturalistic data like this more difficult to understand, which has been shown to cause a significant effect on tone categorization with human listeners in several tone identification studies (Francis, Ciocca, Ma, & Fenn, 2008; Abramson, 1986; Ramadoss, 2011).

Experiment Three used only citation stimuli in training and testing, and *Experiment Four* used a combination of citation and continuous stimuli in training and testing. For Experiment Three, 30 tokens from each tone category were used for training, and 25 citation tokens from each tone category were used for testing. In Experiment Four, 25 citation and 5 continuous training tokens from each tone category were used for training, and 20 citation and 5 continuous tokens from each tone category were used for testing. Based on the results from Experiments One and Two, I decided for both experiments to set the quantization parameter at 20 Hz. I was

motivated to do this, since there was evidence that the *coarse-grained* model performed better with level tones. Based on the acoustic analysis of the tonal data in chapter Two, and from the findings from previous research, a continuous context seems to limit the range in F0 for contour tones. The hypothesis tested between the two experiments was that the inclusion of continuous training and testing stimuli would make it more difficult for the computational model in Experiment Four to correctly identify the contour tone categories. This would be due to less variability and distinctiveness in the model for the contour tones.

4.2.3.1 Percent Correct for Experiment Three and Four

Paired-samples *t*-tests were run showing that each experiment had a greater mean percent correct score in the Post-Test than in the Pre-Test. The *t*-statistics and *p*-values are shown in Table 4.12, along with the mean percent correct for each test. This shows that the EM Training significantly improved the model's test scores in both experiments.

Experiment	Pre-test	Post-test	$t(49) =$	p
3	36.9%	54.6%	14.24	< 0.001
4	37.7%	51.2%	11.66	< 0.001

Table 4.12 Mean Performance per Test for Experiment Three and Four

4.2.3.2 Percent Correct ANOVA

I also ran a two-way repeated measures ANOVA between Experiments Three and Four. This analysis was done to test for any main effects due to the difference in stimulus type between these two experiments. Test was the within-subjects factor, and Experiment was the between-subjects factor used for the ANOVA. The factor levels for Test were the Pre-Test and the Post-Test, and the factor levels for the Experiment factor were Experiment Three and Experiment Four. There was no significant main effect of Experiment on percent correct scores, $F(1,98) = 2.40$, $p = 0.125$, but there was a main effect of Test, $F(1,98) = 337.40$, $p < 0.001$. There was also

a significant interaction of Experiment and Test on percent correct scores, $F(1,98) = 6.33$, $p = 0.014$.

Table 4.13 shows the mean percent correct scores between Experiment Three and Four for the Pre-Test and Post-Test.

Test	Experiment 3	Experiment 4	$t(49) =$	p
Pre-test	36.9%	37.7%	-0.79	= 0.433
Post-test	54.6%	51.2%	2.74	= 0.008

Table 4.13 Mean performance for experiment three and four

For the Pre-test, the mean percent correct score for Experiment Three was not significantly different from the mean percent correct scores for Experiment Four. In the Pre-Test neither Experiment's model had yet been trained, so it was expected that there would be no significant difference between the two experiments.

The Post-test mean percent correct score for Experiment Three was significantly greater than for Experiment Four. The worse mean percent correct scores for Experiment Four was likely due to the effect of combining citation and continuous training and testing stimuli. For the continuous tokens, it was just harder to learn the differences between them. These tokens had less certain information regarding changes in fundamental frequency, since they were suppressed and more clipped[23]. The addition of the continuous tokens to testing and training made the model not learn as successfully in Experiment Four as it did in Experiment Three. These results

[23] It's worth considering how much more variability was introduced into the stimuli by just the different set of recordings, under a different set of conditions. Perhaps having just a learning condition where the model was presented with **only** continuous tokens would be a handy way of testing if there was any effect due to differences in recording conditions of stimuli. However, due to the low number of native Thai speakers recruited for this study, gaining a large database of continuous tokens for use in the model proved difficult.

were not surprising, since I expected the continuous tokens to be more difficult for the model than the citation tokens.

There was a missed opportunity here. The differences in the context type should have also been tested with a greater quantization factor. The fine-grained model should have been better than the coarse-grained model, since the fine-grained model allowed for the storage of more information, which should be true even with different the context types. Even though Experiments One and Two showed no significant differences in the Post-Tests, this is not necessarily true for Experiments Three and Four.

4.2.3.3 Experiment Three: Tone Categorization for Citation-only stimuli

Table 4.14 and Table 4.15 show the confusion matrices of the results of the Pre-Test and Post-Test for Experiment Three. This experiment only used citation stimuli for both tests.

	H	M	L	R	F	Total	Percent Correct
H	320	280	47	92	511	1250	25.6%
M	321	548	139	156	86	1250	43.8%
L	223	290	417	268	52	1250	33.4%
R	151	211	228	601	59	1250	48.1%
F	214	165	174	280	417	1250	33.4%
Total	1229	1494	1005	1397	1125	6250	

Table 4.14 Experiment Three Pre-Test

	H	M	L	R	F	Total	Percent Correct	Change
H	864	123	30	37	196	1250	69.1%	43.5%
M	140	693	277	64	76	1250	55.4%	11.6%
L	59	436	565	105	85	1250	45.2%	11.8%
R	103	221	174	579	173	1250	46.3%	1.8%
F	362	41	31	106	710	1250	56.8%	23.4%
Total	1528	1514	1077	891	1240	6250		

Table 4.15 Experiment Three Post-Test

As with the previous experiments, D-prime measures and response bias scores were calculated. I ran a two-way repeated-measures ANOVA to test the effect of Tone Category and Test Type on the D-prime scores for all simulations in the experiment. There was a significant

main effect of Category, $F(4,196) = 10.45$, $p < 0.001$, and there was also a significant main effect of Test Type, $F(1,49) = 207.70$, $p < 0.001$. There was a significant interaction of Test and Category, $F(4,196) = 30.20$, $p < 0.001$. I also ran a two-way repeated measures ANOVA to test for any main effects on response bias. There was a significant main effect of Category on response bias, $F(4,196) = 9.74$, $p < 0.001$, and there was also a significant main effect of Test, $F(1,49) = 129.10$, $p < 0.001$. There was a significant interaction of Category and Test, $F(4,196) = 13.87$, $p < 0.001$.

Table 4.16 shows the D-prime and response bias scores for each tone category in the Pre-Test and Post-Test.

	D-prime				Bias			
Lexical Tones	Pre-Test	Post-Test	$t(49) =$	p	Pre-Test	Post-Test	$t(49) =$	p
High	0.183	1.678	-19.92	< 0.001	0.896	0.306	7.43	< 0.001
Middle	0.750	1.152	-5.10	< 0.001	0.552	0.428	1.90	= 0.063
Low	0.783	1.240	-5.80	< 0.001	0.915	0.754	1.78	= 0.082
Rising	1.040	1.579	-4.36	< 0.001	0.603	0.906	-3.42	= 0.001
Falling	0.643	1.487	-10.00	< 0.001	0.770	0.563	3.54	= 0.001

Table 4.16 D-prime and bias measures for Experiment Three

In the Post-Test, the model was significantly more biased against the Rising category than it was in the Pre-Test, and for the High and Falling categories, the model was significantly less biased against these categories than in the Post-Test. Falling and Rising are both contour tone categories where fundamental frequency changes over time in a token's utterance, but the direction of the change is opposite each other. The model treats these two contour tones differently. With more information about the fundamental frequency from training stimuli, the model identified all categories better. Tokens produced in a citation context have less variability and the speakers tend to produce the token's tone more accurately. This apparently makes it easier for the model to discriminate between tone categories.

4.2.3.4 Experiment Four: Tone Categorization with combined stimuli

Confusion matrices for both the Pre-Test and Post-Test for Experiment Four are shown in Table 4.17 and Table 4.18. The model in Experiment Four was trained and tested on the combination of citation and continuous stimuli. D-prime measures and response bias scores were also calculated using these tables, and I ran a two-way repeated measures ANOVA with the same factors and levels as per the previous experiment. The analysis was used to test for any significant main effects of Test or Category on D-prime and response bias scores.

	H	M	L	R	F	Total	Percent Correct
H	282	379	30	101	458	1250	22.6%
M	283	496	173	149	149	1250	39.7%
L	194	248	484	272	52	1250	38.7%
R	133	163	270	614	70	1250	49.1%
F	175	178	153	262	482	1250	38.6%
Total	1067	1464	1110	1398	1211	6250	

Table 4.17 Experiment Four Pre-Test

	H	M	L	R	F	Total	Percent Correct	Change
H	655	181	44	39	331	1250	52.4%	29.8%
M	256	585	239	60	110	1250	46.8%	7.1%
L	85	361	594	144	66	1250	47.5%	8.8%
R	95	236	216	567	136	1250	45.4%	-3.7%
F	249	71	31	101	798	1250	63.8%	25.2%
Total	1340	1434	1124	911	1441	6250		

Table 4.18 Experiment Four Post-Test

There was a significant main effect of Category on D-prime measures, $F(4,196) = 29.85$, $p < 0.001$, and there was a significant main effect of Test, $F(1,49) = 137.70$, $p < 0.001$. There was also a significant interaction of Category and Test on D-prime measures, $F(4,196) = 11.63$, $p < 0.001$. For the response bias measures, there was also a significant main effect of Category, $F(4,196) = 7.47$, $p < 0.001$, and a significant main effect of Test, $F(1,49) = 63.98$, $p < 0.001$, and a significant interaction of Category and Test, $F(4,196) = 15.03$, $p < 0.001$.

	D-prime				Bias			
Lexical Tones	Pre-Test	Post-Test	$t(49) =$	p	Pre-Test	Post-Test	$t(49) =$	p
High	0.027	1.193	-8.71	< 0.001	1.077	0.531	5.80	< 0.001
Middle	0.555	0.905	-3.88	< 0.001	0.632	0.541	0.96	= 0.341
Low	0.922	1.274	-4.00	< 0.001	0.796	0.705	1.20	= 0.235
Rising	1.013	1.536	-4.24	< 0.001	0.543	0.893	-5.42	< 0.001
Falling	0.762	1.559	-9.36	< 0.001	0.686	0.387	5.71	< 0.001

Table 4.19 D-prime and Bias scores for Categories in Experiment Four

Table 4.19 shows the mean D-prime and response bias scores for each category in each test. The mean D-prime scores for all categories were significantly greater in the Post-Test than in the Pre-Test. The model was significantly more biased against the Rising category in the Post-Test than it was in the Pre-Test, and the model was significantly less biased against the High and Falling categories in the Post-Test. The model in Experiment Four shows the exact same pattern of sensitivity and response bias that it showed in the previous experiment.

4.2.3.5 D-prime measures between Experiments Three and Four

I compared the D-prime scores of the model between these experiments to test what effect stimulus type (or context) had on the model's ability to categorize tone. I ran a Three-way repeated measures ANOVA to test the effects of Experiment, Tone Category and Test Type on the D-prime scores. I used the same factors and levels as in the previous three-way ANOVA. There was a significant main effect of Experiment, $F(1,98) = 4.45$, $p = 0.037$. There was a significant main effect of Category, $F(4,392) = 35.55$, $p < 0.001$, and there was a significant interaction of Experiment and Category, $F(4,392) = 6.91$, $p < 0.001$. There was a significant main effect of Test, $F(1,98) = 340$, $p < 0.001$, but there was no significant interaction of Test and Experiment, $F(1,98) = 2.14$, $p = 0.147$. There was a significant interaction of Category and Test, $F(4,392) = 36.97$, $p < 0.001$, but there was no significant interaction of Category, Test and Experiment, $F(4,392) = 0.94$, $p = 0.443$.

The lack of significant interactions here was unexpected. I expected there to be a significant difference between the contour categories (Rising and Falling) in both experiments, where Experiment Four would have been less sensitive to these categories due to less information about their contour, duration and shape. However, no statistical difference in D-prime measures emerged from the data. The suspected culprit for this may have been the low number of continuous tokens used in Experiment Four's model. Using only continuous tokens for testing and training the model in Experiment Four could possibly have remedied this situation.

4.2.3.6 Response bias measures between Experiments

I also ran a three-way repeated measures ANOVA to test the effects of Tone Category, Test Type and Experiment on response bias measures in Experiments Three and Four. There was not a significant main effect of Experiment, $F(1,98) = 0.86$, $p = 0.355$. There was a significant main effect of Category, $F(4,392) = 11.99$, $p < 0.001$, and there was a significant interaction of Experiment and Category, $F(4,392) = 5.30$, $p < 0.001$. There was a significant main effect of Test, $F(1,98) = 178.65$, $p < 0.001$, but there was not a significant interaction of Experiment and Test, $F(1,98) = 0.87$, $p = 0.353$. There was a significant interaction of Category and Test, $F(4,392) = 28.60$, $p < 0.001$, but there was not a significant interaction of Category, Test and Experiment, $F(4,392) = 0.29$, $p = 0.886$.

Testing the response bias measures also failed to show any significant interactions between the experiments.

4.3 Discussion of Experiments Three and Four

These experiments failed to produce significant results in testing the hypothesis of whether the continuous context would have a significant effect on identifying the contour tone categories.

However, the discussion presented in Chapter Two and previous studies show that the effects the continuous context has on contour tones is sound (Potisuk, Gandour, & Harper, 1997; Morén & Zsiga, 2006; Zsiga & Nitisaroj, 2007).

It is hard to determine exactly what factors caused there to be no significant interactions between my experiments, but one obvious reason was due to the low number of continuous tokens used in Experiment Four. The citation tokens that were combined with the continuous tokens in this experiment made it so that the two experiments were not significantly different enough. This could have been remedied by including more continuous tokens and less citation tokens in Experiment Four. In fact, the best scenario would have been to use only continuous tokens in Experiment Four, and thus I would have more clearly and more accurately tested my original hypothesis.

The trained models in Experiments Three and Four were tested on different sets of tokens. This is not normally done in computational science, where models are generally tested on the exact same set of tokens (Tungthangthum, 1998). So in a sense, this is comparing apples to oranges, because the models were being evaluated on different sets of tokens. However, the goal in this work was to test how the model performed on tokens with certain properties, not specific tokens per se. This is more in line with perceptual training studies (Laphasradakul, 2010; Zsiga & Nitisaroj, 2007) and also reflects the fact that linguistic knowledge should be generalizable across different sets of tokens.

4.4 Addendum to Continuous vs. Citation

The experiments described in section 4.2.3 and their results failed to show a strong effect between the two experiments, nor did they show a significant interaction between the

experimental test factors. Thus, the results of these experiments did not warrant any conclusion as to the effects that the continuous tokens had on the computational model.

As suggested by some of the reviewers of this dissertation, a new—yet short—experiment was conducted. The analysis of the continuous data in Chapter Two, and also the review of the literature support the claim that tones produced in a continuous context are more difficult for tone perception than when they are produced in citation context. This new experiment aims to support this claim.

I compared two groups, each with 25 trials. The *first group* included only five citation tokens from each tone category, and the *second group* included only five continuous tokens from each tone category. I constrained both groups to select tokens from the same list of words, with the only difference between the two lists being that one was produced in a citation context and the other was produced in a continuous context. I ran a repeated measures 2-way ANOVA with Test and Group as factors. Test had two levels: Pre-Test and Post-Test, and Group had two levels: Citation group and Continuous group. The interaction of Test and Group had a significant effect on performance scores, $F(1,48) = 5.82$, $p = 0.020$. A paired-samples t-test showed that the citation group's performance ($M = 61.1\%$) significantly outperformed the continuous group's performance ($M = 46.6\%$) in the post-test, $t(48) = 5.38$, $p < 0.001$.

I also ran a repeated-measures 3-way ANOVA to test for any effects of Category, Test and Group on D-prime scores. The interaction of Category, Test and Group had a significant effect, $F(4,240) = 3.71$, $p = 0.001$. I ran t-tests on the post-test D-prime scores of each category between the two experimental groups. The citation group's mean D-prime score for the Mid category ($M = 1.610$) was significantly greater than the continuous group's D-prime score for the Mid category ($M = -0.003$), $t(48) = 4.69$, $p < 0.001$. The citation group's mean D-prime score for

the Low category ($M = 2.827$) was significantly greater than the continuous group's mean D-prime score for the Low category ($M = 0.655$), $t(48) = 7.19$, $p < 0.001$. The citation group's mean D-prime score for the Rising category ($M = 2.384$) was significantly greater than the continuous group's mean D-prime score for the Rising category ($M = 1.374$), $t(48) = 2.08$, $p = 0.043$.

The model used here was essentially the same as in the previous experiments, but the tokens tested and trained in the first experimental group were all citation and in the second experimental group they were all continuous. As a linguist, what is important to me is testing for a significant difference between the citation and continuous tokens in some experimental method. I chose to use a HMM to test this, but the dissertation's experimental model failed to test this with any significance. I believe this new, simpler experiment supports my original conclusion in that the citation and continuous tokens are different in terms of tone perception. However, further tests and experiments are needed to draw deeper conclusions than the ones presented here, but that will be saved for future endeavors.

Chapter Five: Human Listening Experiments

A perceptual experiment was run to test whether non-Thai speakers could learn to identify the lexical tone categories of Thai words from audio recordings. The set of test and training stimuli used in this tone identification experiment was the same set used in the testing of the computational model in chapter four. The multi-speaker database was used for the tone identification task with the human listeners. The intent was to match everything up between the human and the computational tests, in order to compare the performance of the computational model to the performance of human listeners in an actual experiment. This was done to see if the computational model could perform at the same level or better than the listeners did in the tone identification experiment, and also just to see where the discrepancies in the two patterns of behavior were.

For the tone identification experiments there were two groups of listeners. Each group had 15 participants. The first group was tested and trained with only citation stimuli. The second group was tested and trained on a combination of citation and continuous stimuli. All stimuli tokens for both experiments were taken from the multi-speaker database of tokens. I also chose to use the multi-speaker database, because tone identification is not just limited to a single speaker but to multiple speakers.

5.1 Tone Identification Experiment

Figure 5.1 shows the framework for the identification experiment. The experimental framework begins with the *Pre-Test Phase*, followed by the *Training Phase* and then the *Post-Test Phase*.

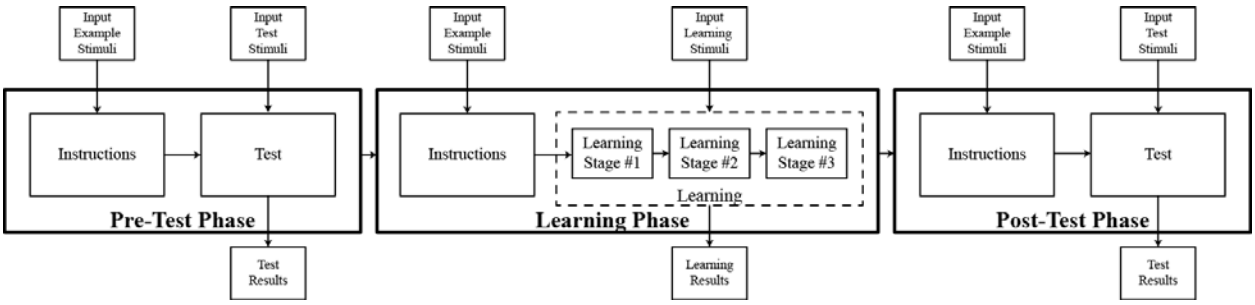

Figure 5.1 Identification Experiment

The *Pre-Test Phase* included two input operations, two components and an output operation. *Instructions* was the first component of this phase, where listeners were instructed on how to do the experiment. Subjects listened to five example stimuli from each of the five lexical tone categories, in order to familiarize themselves with the tone categories. The "category" labels were meant to be non-descriptive, so as to not give the listeners any added advantage that the computational model would not have. Figure 5.2 is a screen shot of the instructions panel for one of the example stimulus categories. There was a separate instruction screen for each of the five categories.

Figure 5.2 Example Participant Instructions

After the subjects listened to an example stimuli set for each tone category, they proceeded to the *Test* component of the Pre-Test. The *Input Test Stimuli* operation randomly selected 125 test stimuli from the database, with 25 tokens for each lexical tone category. These stimuli were presented to the listeners in a tone category identification task.

Figure 5.3 is a screen shot for a test trial during the *Test Stage* of the Pre-Test Phase. During the trial, subjects categorized each test stimulus into one of five tone categories. In Figure 5.3, the meaning of "Choice" for the listener equals the question: what category did the stimulus belong to?

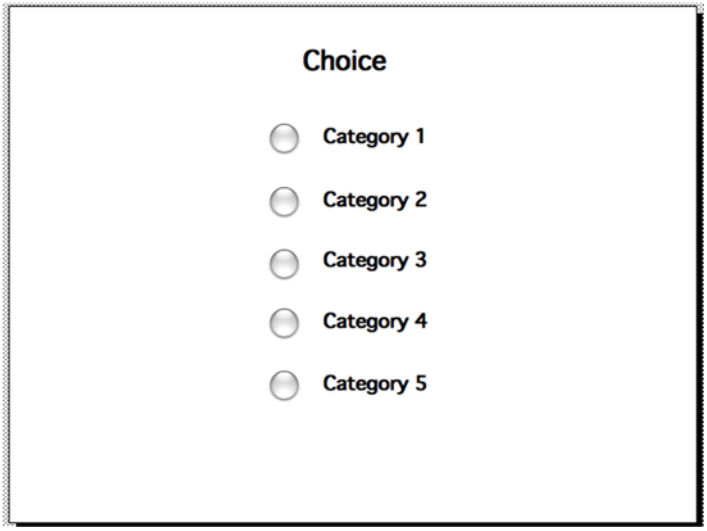

Figure 5.3 Pre-test example

The traditional Thai labels were not presented to the subjects during the experiment. Instead, the labels "Category 1, Category 2, . . . , Category 5" were used instead. I chose to go with this method was because the computer would not have *a priori* category expectations the way the listeners might have had, if they had been given descriptive tone category labels. The computational model would not have the advantage of altering its expectations about the tone category content based simply on the names of the categories alone. However, the design

decision proved to be challenging for listeners, because it might have caused some listeners to not focus directly on the F0 dimension of the stimuli, which was something that the HMM model was explicitly designed to do[24].

The *Test Results* operation recorded both the correct category for each stimulus, and the subject's response to that stimulus. If the subject's response matched the correct category for the test stimulus, then the subject scored a hit. These results were scored for all 125 test stimuli for each subject.

In the *Learning Session*, subjects began with reading the instructions and reviewing the same five example stimuli presented in the Pre-Test session. The instructions explained the process to follow in the three *Learning* stages. Figure 5.4 is a screen shot from the learning session.

[24] However, within the instructions listeners were informed that Thai is a tone language and it has five tone categories. A snippet of the instructions from Pre-Test stated ". . . Thai is a tone language, where the pitch (much like a musical note in singing or played on an instrument) determines the meaning of a word. Thai has 5 unique tone categories. . . ." Even though listeners were not explicitly directed to listen for falling or rising pitch contours, they may been aware of this at some level.

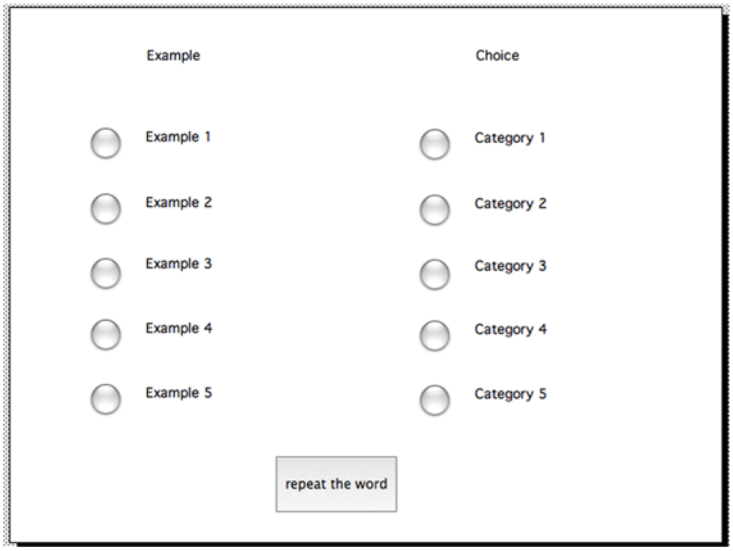

Figure 5.4 Learning session

Each Learning Stage had 50 trials, for a total of 150 trials in the Learning Phase. In *Learning Stage #1*, participants were able to re-play the stimulus for the current learning trial by clicking on a "repeat the word" button (see Figure 5.4). This option was given to them prior to making their category selection for the current stimulus. Participants in this stage were also able to repeat examples of each category prior to making their decision. These were the same example stimuli that had been given during the *instructions stage* of the learning phase. Finally, the participants were shown the correct tone category after making a decision for the given stimulus in each trial. In *Learning Stage #2*, participants were no longer able to re-play the stimulus for the current learning trial. They could still repeat examples corresponding to each tone category, and they were still shown the correct tone category after making their decision. In *Learning Stage #3*, participants were only shown the correct tone category after making their decision. Table 5.1 shows the learning aids made available in each stage.

Learning Session Options			
Aids	Stage 1	Stage 2	Stage 3
Correct Answer	●	●	●
Repeat Examples	●	●	
Repeat Stimulus	●		

Table 5.1 Learning Options

The goal of using this step-wise process was to get the participants to become familiar with the process used in the testing, while still training them to identify the correct tone category in a short amount of time.

The *Learning Results* were scored in the same manner as the Pre-Test results. Information was also collected for how many times a subject clicked the button to repeat the stimulus word in *Learning Stage #1* for the first 50 trials. Likewise, information was collected for how many times that the subject listened to example stimuli for each category in learning stages one and two.

The *Post-Test Session* was identical to the Pre-Test session. Subjects reviewed the same instructions and example stimuli, and performed the same tone category identification task. Figure 5.3 is also a representative screen shot of a Post-Test trial. A different set of 125 test stimuli were randomly selected for the Post-Test, with 25 stimuli from each lexical tone category. Finally, *Test Results* were scored in the same manner as the test results from the Pre-Test session.

5.2 Results

Identification accuracy scores for each participant were calculated in the same manner as in the experiments that were run on the computational model in chapter four. Accuracy was calculated as the total number of correct hits over the total number of trials. I ran a two-way

repeated-measures ANOVA to test the effects of Experiment and Test on identification accuracy scores. Test was a within-subjects factor. Experiment was a between-subjects factor, and both experiments were its levels. There was not a significant main effect of Experiment, $F(1,28) = 0.53$, $p = 0.475$, but there was a significant main effect of Test, $F(1,28) = 5.91$, $p = 0.022$. There was not a significant interaction of Experiment and Test, $F(1,28) = 0.35$, $p = 0.558$. Overall there was an improvement in accuracy from the Pre-Test to the Post-Test. The combined mean identification accuracy score for the Pre-Test from both experiments was 23.2%, and the combined mean score from the Post-Test was 27.7%. Using a paired-samples *t*-test, the combined Post-Test score was significantly greater than the Pre-Test, $t(29) = 2.46$, $p = 0.020$. Performance was effectively at chance during the Pre-Test in this experiment—26.4% was the upper boundary for a score to be considered greater than chance. For both experiments, the mean Post-Test scores were above chance. Both experiments proved difficult for the listeners, and this was in part due to the difficulty of them listening to stimuli from the multi-speaker database.

5.2.1 Analysis based on category

Confusion matrices for Experiment Five are presented in Table 5.2 for the Pre-test results, and in Table 5.3 for the Post-test. The vertical columns represent the listener's responses and the horizontal rows represent the tone category of the test stimuli. The percent correct for each test is also shown, and in the Post-Test the change in the percent correct value from the Pre-Test to the Post-Test is also given. Along the bottom is given the percentage of responses for each category for each test.

	H	M	L	R	F	Total	Percent Correct
H	58	103	83	61	70	375	15.5%
M	82	83	81	52	77	375	22.1%
L	45	87	102	53	88	375	27.2%
R	94	67	71	68	75	375	18.1%
F	58	79	77	62	99	375	26.4%
Total	337	419	414	296	409	1875	
Percent Response	18.0%	22.3%	22.0%	15.8%	21.8%		

Table 5.2 Pre-Test confusion matrix for Experiment Five

	H	M	L	R	F	Total	Percent Correct	Change
H	80	60	153	50	32	375	21.3%	5.8%
M	85	117	34	50	89	375	31.2%	9.1%
L	46	106	128	31	64	375	34.1%	6.9%
R	87	74	24	110	80	375	29.3%	11.2%
F	84	89	58	62	82	375	21.9%	-4.5%
Total	382	446	397	303	347	1875		
Percent Response	20.4%	23.8%	21.2%	16.2%	18.5%			

Table 5.3 Post-Test confusion matrix for Experiment Five

For Experiment Six, Table 5.4 shows the confusion matrix for the Pre-Test, and Table 5.5 shows the results for the Post-Test.

	H	M	L	R	F	Total	Percent Correct
H	88	88	102	41	60	379	23.2%
M	66	77	63	66	97	369	20.9%
L	45	87	103	43	78	356	28.9%
R	76	80	47	107	81	391	27.4%
F	78	90	79	50	83	380	21.8%
Total	353	422	394	307	399	1875	
Percent Response	18.8%	22.5%	21.0%	16.4%	21.3%		

Table 5.4 Pre-Test confusion matrix for Experiment Six

	H	M	L	R	F	Total	Percent Correct	Change
H	76	58	142	43	49	368	20.7%	-2.5%
M	57	81	96	47	102	383	21.1%	0.2%
L	37	92	160	29	56	374	42.8%	13.9%
R	96	68	46	95	72	377	25.2%	-2.2%
F	54	100	77	31	111	373	29.8%	8.0%
Total	320	399	521	245	390	1875		
Percent Response	17.1%	21.3%	27.8%	13.1%	20.8%			

Table 5.5 Post-Test confusion matrix for Experiment Six

Percent correct, D-prime and response bias scores were calculated for each category in each experiment and test, in the same manner as for the computational model in chapter four.

5.2.1.1 Experiment Five: Category Analysis

In Experiment Five, the listeners were only presented with only citation stimuli for testing and training. I ran a two-way ANOVA to test the effects of Category and Test on D-prime and on response bias scores. Category had five levels, and Test had two levels. Both Category and Test were within-subjects factors. For D-prime scores, there was a significant effect of Category, $F(4,56) = 3.77$, $p = 0.009$, and there was also a significant effect of Test, $F(1,14) = 5.31$, $p = 0.037$. However, there was not a significant interaction of Category and Test, $F(4,56) = 1.32$, $p = 0.274$. The Post-Test mean D-prime score (0.285) was significantly greater than the Pre-Test mean D-prime score (-0.007), $t(74) = 2.92$, $p = 0.005$.

For response bias, I ran a similar two-way ANOVA. There was a significant main effect of Category, $F(4,56) = 75.04$, $p < 0.001$, and there was also a significant main effect of Test, $F(1,14) = 6.36$, $p = 0.024$. There was, however, no significant interaction of Test and Category, $F(4,56) = 0.57$, $p = 0.688$. The Pre-Test mean response bias (0.715) was significantly greater than the Post-Test mean response bias (0.612), $t(74) = 2.55$, $p = 0.013$.

5.2.1.2 Experiment Six: Category Analysis

The listeners heard the combined citation and continuous stimuli in this experiment. I ran two more Two-way ANOVAs on the D-prime and response bias scores from Experiment Six. These ANOVAs had the same structure as the ones for Experiment Five. For the D-prime scores, there was a significant main effect of Category, $F(4,56) = 6.68$, $p < 0.001$. However, there was not a significant main effect of Test, $F(1,14) = 1.88$, $p = 0.192$, nor was there a

significant interaction of Category and Test, $F(4,56) = 1.10$, $p = 0.367$. For these listeners there was no significant improvement from the Pre-Test to the Post-Test.

For response bias scores, there was a significant main effect of Category, $F(4,56) = 61.98$, $p < 0.001$, but there was no significant main effect of Test, $F(1,14) = 1.55$, $p = 0.233$. However, there was a significant interaction of Category and Test, $F(4,56) = 3.60$, $p = 0.007$. Table 5.6 shows the mean response bias for each tone category in each test. This table also lists the results of a Paired-Samples *t*-test run between the bias scores from each test. The Low category was the only category to show a significant change in response bias scores.

	Bias			
Lexical Tones	Pre-Test	Post-Test	$t(49) =$	p
High	0.856	0.955	-1.15	= 0.269
Mid	0.813	0.848	-0.37	= 0.716
Low	0.805	0.465	2.30	= 0.037
Rising	0.910	1.019	-1.92	= 0.076
Falling	0.002	-0.174	2.08	= 0.057

Table 5.6 Experiment Six response bias scores between tests per category

It is important to point out that the listeners did produce a lot of correct responses to the Low tone category, so perhaps it was the easiest one for them to identify correctly. What is interesting here is that the experiments were set up to test listener's sensitivity, or lack thereof, to the continuous stimuli. As shown in Chapter Two, these stimuli tend to have less F0 frequency range, are shorter and somewhat clipped. This was particularly the case for the contour tones. However, it seems that the listeners in this experiment significantly increased in favor of the Low category stimuli in the Post-Test than they were in the Pre-Test.

5.2.1.3 Experiments Five and Six

I wanted to test if the listeners in Experiment Six were affected differently by the combined stimuli. To test this, I ran a Three-way repeated-measures ANOVA on the D-prime and response bias scores from both experiments. Category and Test were the within-measures

factors and they were treated the same as they were in the previous ANOVA. Experiment was the between-subjects factor, with each experiment as a level.

For D-prime measures, there was not a significant main effect of Experiment, $F(1,28) = 0.92$, $p = 0.347$. There was a significant main effect of Category, $F(4,112) = 8.69$, $p < 0.001$, but there was no significant interaction of Experiment and Category, $F(4,112) = 1.91$, $p = 0.113$. There was a significant main effect of Test, $F(1,28) = 6.84$, $p = 0.014$, but there was no significant interaction of Experiment and Test, $F(1,28) = 0.52$, $p = 0.476$. There was not a significant interaction of Category and Test, $F(4,112) = 1.20$, $p = 0.314$, nor a significant three-way interaction of Experiment, Category and Test, $F(4,112) = 1.26$, $p = 0.292$. The combined stimuli did not have any significant effect on listeners' sensitivity to the test and training stimuli between the two experiments.

For response bias scores, there was not a significant main effect of Experiment, $F(1,28) = 0.19$, $p = 0.666$. There was a significant main effect of Category, $F(4,112) = 134.39$, $p < 0.001$, but there was no significant interaction of Experiment and Category, $F(4,112) = 1.07$, $p = 0.376$. There was a significant main effect of Test, $F(1,28) = 6.93$, $p = 0.014$, but there was no significant interaction of Experiment and Test, $F(1,28) = 0.66$, $p = 0.425$. There was also no significant interaction of Category and Test, $F(4,112) = 1.81$, $p = 0.133$, but there was a significant three-way interaction of Experiment, Category and Test on response bias scores, $F(4,112) = 2.98$, $p = 0.022$.

Bias

	Pre-Test				Post-Test			
Categories	Experiment 5	Experiment 6	$t(49) =$	p	Experiment 5	Experiment 6	$t(49) =$	p
High	1.010	0.856	1.85	= 0.086	0.876	0.955	-0.88	= 0.396
Mid	0.801	0.813	-0.18	= 0.858	0.659	0.848	-3.46	= 0.004
Low	0.805	0.805	0.00	= 0.997	0.679	0.465	1.93	= 0.074
Rising	1.029	0.910	1.40	= 0.185	0.899	1.019	-1.29	= 0.218
Falling	-0.070	0.002	-0.83	= 0.423	-0.050	-0.174	1.30	= 0.215

Table 5.7 Bias scores per category between experiments

The response bias scores are broken down in Table 5.7 for each category and test in each experiment. In the Pre-Test there were no significant differences, but in the Post-Test, the mean response bias score for the Mid category was significantly lower in Experiment Five than in Experiment Six.

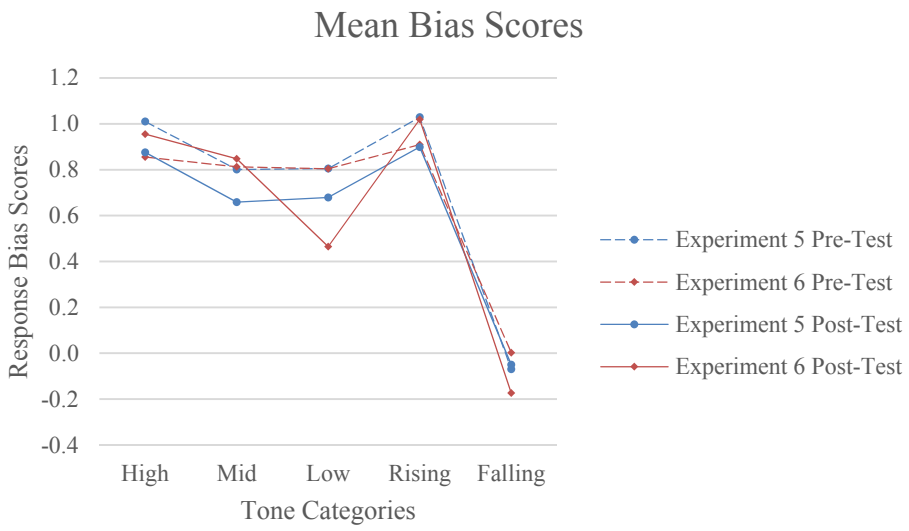

Figure 5.5 Comparison of mean bias scores per category

Figure 5.5 shows the comparison of mean bias scores for each category in each test. The falling response bias scores in all four tests probably account for the three way interaction. Based on these scores, listeners in both experiments were highly in favor of responding with the falling category over the other categories. The reason this happened was because the "Falling" category button was closest to the "Play next word" button in the layout of the experiment on two computer screens.

5.3 Learning

This section compares the listeners' performance in the different stages in the Learning session.

5.3.1 Overall performance

I wanted to test if learning stage had an effect on listeners' performance in the task in each experiment. I ran a one-way ANOVA to test the effect of Learning Stages on identification accuracy scores. Learning Stage was the within-subjects factor, and it had three levels. In Experiment Five, there was a significant effect of Learning Stage on identification accuracy, $F(2,28) = 3.97$, $p = 0.030$. Post-hoc tests did not reveal a significant increase in the mean accuracy scores from Stage One (29.7%) to Stage Two (31.1%), $t(14) = 0.79$, $p = 0.44$, but there was a significant increase in scores from Stage Two to Stage Three (35.3%), $t(14) = 2.35$, $p = 0.034$.

For Experiment Six, I ran a similar ANOVA to test for any significant effect of Learning Stage on accuracy scores. However, there was not a significant main effect of Stage, $F(2,28) = 0.45$, $p = 0.64$.

I ran a Two-way repeated-measures ANOVA to test the effects of Stage and Experiment on identification accuracy scores. Learning Stage was the within-subjects factor, and Experiment was the between-subjects factor, with two levels. There was not a significant main effect of Experiment, $F(1,28) = 0.71$, $p = 0.408$, and there was not a significant effect of Stage either, $F(2,56) = 0.35$, $p = 0.706$. Likewise, there was not a significant interaction of Experiment and Stage, $F(2,56) = 2.86$, $p = 0.065$.

Between experiments there was a nearly significant divergence in performance scores at Stage Three. There was more consistent improvement in Experiment Five. Table 5.8 shows the comparison between Experiments for each learning stage.

	Experiment Five	Experiment Six	$t(28) =$	P
Stage 1	29.7%	30.7%	-0.28	= 0.779
Stage 2	31.1%	30.1%	0.25	= 0.805
Stage 3	35.3%	28.0%	2.00	= 0.056

Table 5.8 Accuracy between experiments

There is no significant difference between experiments in Stages One and Two. However, for Stage Three, Experiment Five's mean identification accuracy is almost significantly higher than Experiment Six's ($p = 0.056$). Figure 5.6 shows these values compared with each other on a graph.

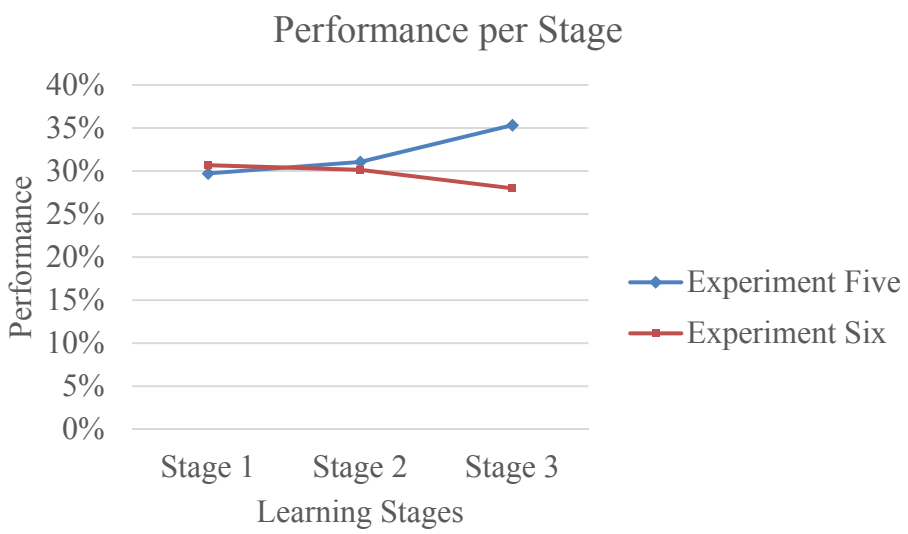

Figure 5.6 Performance between Experiment Five and Six

During the Learning Phase, both subject groups showed no significant improvement in performance between Stage One and Stage Two. However in Stage Three, there was learning in

Experiment Five even though there was no learning in Experiment Six. Experiment Five's listeners almost performed significantly better than Experiment Six's at this Stage in the session.

5.4 Discussion

The listeners improved less in tone identification than the simulations run with the computational model. The lack of significant differences between tests and experiments may result from the small number of learning sessions in between the Pre-Test and Post-Test. Including more learning sessions would likely yield greater increases in tone identification accuracy from the Pre-Test to the Post-Test. Another factor that could be tested in the future would be to include another subject group that would be given the traditional labels for the lexical tone categories in Thai. Perhaps in the present experiments the lack of traditional tone labels made learning the relevant differences between tone categories too difficult of a task for naïve listeners. Another way to test this difficulty would be to have a group of native Thai listeners. Testing the performance of Thai listeners would provide insight into the upper limit of performance on the tone classification task by human listeners. It would provide more context for evaluating the performance of the HMM.

Chapter Six: Conclusion

6.1 Remarks

In this dissertation I set out to do two things. The first was to create and test a computational model for the purpose of tone categorization, and the second was to conduct a tone perception task with naïve listeners of Thai. The primary investigation in this dissertation was to determine if the model could accurately categorize Thai tones. I also tested the ability of listeners to identify tone categories. I did this in order to compare the model to the human listeners to see if it would have a similar pattern of performance. In particular, I intended to investigate if the continuous context would have the same or similar effects on both the model's and the human listener's performance.

6.1.1 Quantization Factor

I conducted the first two experiments in order to test if there was an effect on performance when changing the quantization factor of the model. In the first experiment, the model was considered to be *coarse-grained*, and its quantization parameter was set at $20\ Hz$. In the second, the model was considered to be *fine-grained*, and its parameter was set at $10\ Hz$. In order to reduce the amount of variability in the model, tokens from the single speaker database were used for testing and training.

The results of these two experiments were compared with each other. The fine-grained model had a significantly greater performance in the Pre-Test than the coarse-grained model, but this advantage disappeared in the Post-Test. The fine-grained model had an advantage when there was limited acoustic information derived only from a small set of example tokens.

Typically, more acoustic information would entail the model performing better. This would naturally follow from the precepts of Exemplar Theory. In Exemplar Theory, people store

in memory detailed instances of a perceptual category as acoustic information (such as an instance of a lexical tone category) as it occurs naturally (Johnson, 2007; Nosofsky, 1986; Nosofsky, 1988; 1991). However, it is possible that more information can also make it harder to generalize patterns. This was the case for the fine-grained model as the increase in acoustic information did not significantly increase its performance in the Post-Test. This is a case of missing the forest for the trees[25] [26].

6.1.2 Combined Stimuli

Another set of experiments tested if more naturalistic data would affect the model's identification accuracy of Thai tones. The first of these experiments used only citation tokens from the multiple speaker database for the purpose of testing and training. The second used a combination of citation and continuous tokens from the same database. However, there was no significant difference in the performance of the model between these two experiments.

The continuous tokens used in Experiment Five did not prove to have any significant effect at all. This is perhaps due to the structure of the test: a better comparison would have presented listeners in this experiment with only continuous tokens. For future research this could be remedied by recruiting more speakers, and by increasing the number of carrier sentences that the stimuli tokens were uttered in.

---

[25] The position taken here could be considered to be going against Exemplar Theory, or that it is wrong in this instance. However, in order to be fair, I employed a large amount of normalization of the tones in the model prior to testing the model, and this does not line up (at least in spirit) with an exemplar theoretic model of tone identification.
[26] It may also be the case that the structure of the model forced this particular result of the coarse-grained model performing better than the fine grained model.

6.1.3 Perception Experiment

I also conducted a set of perception experiments to investigate the effect that the combined stimuli had on naïve listeners in a tone identification task. The perception experiments were similar in structure to the computational model. The first experiment acted as a control. The listeners in this experiment were tested and trained only on citation stimuli. The second experiment contained the combined stimuli for testing and training.

Unlike the earlier experiments on the computational model, these experiments did not show any significant difference between them. The combined stimuli did not have a significant effect on listeners' identification accuracy. It is possible that the low number of participants yielded less statistical power when testing for a main effect caused by the combined stimuli.

There was some evidence where the listeners in the control experiment performed better during the last stage of the Learning Phase. Perhaps an increase in the number of these phases would have had a significant effect.

The computational model in both sets of experiments was able to perform significantly better than the listeners did in their experiments. One reason the model performed better was that the tokens presented were normalized for pitch. This greatly reduced the level of variability within the tokens presented to the model. A set of normalized tokens—or just tokens from one speaker alone—would have reduced the amount of variability presented to the listeners, which would make tone categorization easier.

One design element of the perception experiment was the lack of traditional tone category labels during the tone identification task. I did not want the labels to allow listeners to make *a priori* distinctions about the nature of the tone categories. Instead, I wanted listeners to rely solely on acoustic information to make categorical distinctions, and not the "prototype"

information they might get from labels. This obviously proved to be a difficult task, especially with just one Learning Phase. However, based on the apparent difficulty the listeners had when they could not rely on traditional labels, the lack thereof may have caused listeners to be less sensitive to the fundamental frequency of the stimuli presented for testing and learning. The computational model had the advantage of just being presented with the fundamental frequency of each token, whereas the listeners were presented with not just the fundamental frequency, but each token's entire spectral range of frequencies and also its segmental content. On top of that, speakers also had to contend with non-normalized tokens from the multiple speaker database. Listeners were presented with just too much information, along with a very limited categorical information to use to discriminate between tone categories[27]. Just like the first set of experiments run on the computational model, too much acoustic information introduced too much variability for the listeners to learn how to do the tone identification task accurately. Thus, Learning did not significantly increase listener performance on the Post-Test. This was another case of missing the forest for the trees.

6.2 In Conclusion and Onward

From the results of the computational model, F0 is a strong signal for identifying tones. It is possible that focusing the attention of the listeners in on this information would help them more accurately identify tones. More learning and exposure to the acoustic information would be helpful, but also non-acoustic information such as category could improve accuracy. This information has been shown in other tone identification studies to increase performance, and the

[27] This is what native listeners are presented with on an everyday basis. However, native listeners learn how to classify tone categories over many years of experience, while the listeners in this experiment only had one learning session.

lack of such information in the perception experiment in this dissertation supports this notion. This can be tested in future perception experiments where listeners are presented with non-acoustic information, such as a traditional category label for each tone, along with F0 in both testing and training. This information could be presented to one group, and a control group would just be presented with the acoustic information. The performance of each group could be tested to see if the category labels might have a significant effect on learning and on the performance of accurately identifying tone categories. Also, it is necessary to test a control group of native Thai listeners to see how their performance compares to both human listeners and the model, and to see what effects category labels have on native listener tone identification.

Within the experiments run on the model and on the listeners, I attempted to control the amount of variability presented to each. This was fairly easy to do for the computational model's experiments through the use of the quantization factor and normalization of tokens. Yet, the addition of more naturalistic tokens to the model caused it to behave in an unexpected manner. The more naturalistic tokens presented even more variability into the model. The perception experiment with listeners proved to be most problematic, as it was more difficult to control for the amount of variability in the tokens presented to listeners, unless I were to have presented them the single speaker database, too. This last option is something that can be tested in a future experiment on listeners.

References

(2015). Retrieved from thai-language.com: http://www.thai-language.com/dict/

Abramson, A. (1960). *The Vowel and Tones of Standard Thai: Acoustical Measurements and Experiments.* PhD thesis, Columbia University.

Abramson, A. (1986). The Thai Tonal Space. *Status Report on Speech Research, 85*, 105-114.

Baum, L., & Petrie, T. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics, 37*(6), 1554-1563.

Baum, L., Petrie, T., Soules, G., & Weiss, N. (1970). A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics, 41*(1), 164-171.

Boersma, P. (1993). Acurate Short-term Analysis of the Fundamental Frequency and the Harmonic-to-Noise Ratio of a Sample Sound. *Proceedings of the Institute of Phonetic Sciences. 17*, pp. 97-110. Netherlands: University of Amsterdam.

Chao, Y.-R. (1930). A system of tone letters. *La Maître phonétique, 45*, 24-27.

Demeechai, T., & Mäkeläinen, K. (2001). Recognition of syllables in a tone language. *Speech Communication, 33*, 241-254.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum Likeklihodd from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, 39*(1), 1-38.

Francis, A., Ciocca, V., Ma, L., & Fenn, K. (2008). Perceptual Learning of Cantonese Lexical Tones by Tone and Non-Tone Language Speakers. *Journal of Phonetics, 36*, 268-294.

Gandour, J. (1975). On the Representation of Tone in Siamese. In J. G. Harris, & J. Chamberlain (Eds.), *Studies in Tai Linguistics in Honor of William J. Gedney.* Bangkok: Central Institute of English Language, Office of State University, Bangkok.

Johnson, K. (2007). Decisions and Mechanisms in Exemplar-Based Phonology. In M. Sole, P. Beddor, & M. Ohala, *Experimental Approaches to Phonology. In Honor of John Ohala* (pp. 25-40). Oxford: Oxford University Press.

Ladefoged, P., & Johnson, K. (2011). *A Course in Phonetics, Sixth Edition.* Boston: Wadsworth.

Laphasradakul, D. (2010). Training Native Speakers of American English to Perceive Thai Tones Using High Stimulus Variability. University of Florida.

Lehiste, I. (1970). *Suprasegmentals.* Cambridge, Massachusetts: The MIT Press.

Lisker, L., & Abramson, A. (1964). A Cross-Language Study of Voicing in Initial Stops: Acoustical Measurements. *Word, 20*(3), 384-422.

Lobanov, B. (1971). Classification of Russian Vowels Spoken by Different Speakers. *Acoustical Society of America, 49*(2), 606-608.

Manning, C., & Schütze, H. (1999). Markov Models. In *Foundations of Statistical Natural Language Processing.* Cambridge, Massachusetts: The MIT Press.

Morén, B., & Zsiga, E. (2006). The lexical and post-lexical phonology of Thai tones. *Natural Language and Linguistic Theory, 24*, 113-178.

Nitisaroj, R. (2006). Effects of Stress and Speaking Rate on Duration and Tone in Thai. Georgetown University.

Nosofsky, R. (1986). Attention, Similarity, and the Identification-Categorization Relationship. *Journal of Experimental Psychology: General, 115*(1), 39-57.

Nosofsky, R. (1988). Exemplar-Based Accounts of Relations Between Classification, Recognition, and Typicality. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*(4), 700-708.

Nosofsky, R. (1991). Tests of an Exemplar Model for Relating Perceptual Classification and Recognition Memory. *Journal of Experimental Psychology: Human Perception and Performance, 17*(1), 3-27.

Pike, K. (1957). *Tone Languages.* Ann Arbor: University of Michigan Press.

Potisuk, S., Gandour, J., & Harper, M. (1997). Contextual Variations in Trisyllabic Sequences of Thai Tones. *Phonetica, 54*, 22-42.

Qian, Y., & Soong, F. (2009). A Multi-Space Distribution (MSD) and two-stream tone modeling approach to Mandarin speech recognition. *Speech Communication, 51*, 1169-1179.

Rabiner, L. (1989). A Tutorial on Hidden Markove Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE, 77*(2), 257-286.

Ramadoss, D. (2011). The Phonology and Phonetics of Tone Perception. Johns Hopkis University.

Schaefer, V., & Darcy, I. (2014). Lexical function of pitch in the first language shapes cross-linguistic perception of Thai tones. *Laboratory Phonology, 5*(4), 489-522.

Teeranon, P. (2007). The Change of Standard Thai High tone: An Acoustic Study. *Journal of Theoretical Linguistics, 4*(3), 1-17.

Tingsabadh, K., & Abramson, A. (1999). Thai. In *Handbook of the International Phonetic Association* (pp. 147-15-0). United Kingdom: Cambridge University Press.

Tungthangthum, A. (1998). Tone Recognition for Thai. *The 1998 IEEE Asia-Pacific Conference on Circuits and Systems*, (pp. 157-160). Chiangmai.

Wayland, R., & Guion, S. (2003). Perceptual discrimination of Thai tones by naive and experienced learners of Thai. *Applied Psycholinguistics, 24*, 113-129.

Wayland, R., & Li, B. (2008). Effects of Two Training Procedures in Cross-Language Perception of Tones. *Journal of Phonetics, 36*, 250-267.

Wickens, T. (2002). *Elementary Signal Detection Theory.* Oxford: Oxford University Press.

Winters, S. (2010). The long-term retention of fine-grained phonetic detail: evidence from a second language voice identificantion training task. *Talk presented at the 2010 Annual Conference of the Canadian Acoustical Association.* Victoria, British Columbia.

Wu, J., Zahorian, S., & Hu, H. (2013). Tone recognition for continuous accented Mandarin Chinese. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, (pp. 7180-7183). Vancouver, BC.

Yip, M. (2002). *Tone.* Cambridge: Cambridge University Press.

Zsiga, E. (2008). Modeling diachronic change in the Thai tonal space. *U. Penn Working Papers in Linguistics, 14*(1), 395-408.

Zsiga, E., & Nitisaroj, R. (2007). Tone features, tone preception, and peak alignment in Thai. *Language and Speech, 50*(3), 343-383.

APPENDIX A: THAI PHONETICS

A.1. Consonants

Thai exhibits a three-way contrast for voicing consonants. It has voiceless, voiced and aspirated consonants in onset position. The voicing contrasts are primarily cued by Voice Onset Time (VOT) (Lisker & Abramson, 1964)[28], which is "the interval between the release of a closure and the start of the voicing" (Ladefoged & Johnson, 2011, p. 151). This is true for the Thai bilabial and alveolar stops, except for the velar plosives which have no voiced counterpart.

Table 0.1 shows the IPA symbols used for transcribing the Thai consonants. These symbols represent the Standard Central Thai dialect (Tingsabadh & Abramson, 1999).

	Bilabial	Labia-dental	Alveolar	Post-Alveolar	Palatal	Velar	Glottal
Stops	p pʰ b		t tʰ d			k kʰ	ʔ
Nasal	m		n		ŋ		
Fricative		f	s				h
Affricate				tɕ tɕʰ			
Trill			r				
Approximate					j	w	
Lateral- Approximate			l				

Table 0.1 IPA Thai consonants

All consonants can occur in the onset position of the syllable. Only the consonants /p, t, k, ʔ, m, n, ŋ, w, j/ occur in syllable final position. The consonants /p, t, k, ʔ/ have no audible release when they are in the coda. The trill /r/ can also be realized as a phonetic tap [ɾ] (Tingsabadh & Abramson, 1999).

[28] Thai is one of the languages with a 3-way voicing contrast that was used by Lisker and Abramson (1964) in their cross-language study of voicing contrasts.

A.2. Vowels

Thai has nine vowels that are front, central and back. It also has high, mid and low vowels. Table 0.2 shows the nine vowel system for Thai. Thai also has four diphthongs:/ia/, /ɨa/, /ua/, /ai/.

	Front	Central	Back
High	i	ɨ	u
Mid	e	ə	o
Low	æ	a	ɔ

Table 0.2 Thai Vowels

A.3. Syllables

Syllables are either long or short. This is generally realized as either a long vowel (VV) or as a short vowel (V). They may have an onset and a coda, which gives the possible combinations of V, CV, VC, CVC for short syllables, and VV, CVV, VVC, CVVC for long syllables.

APPENDIX B: THAI SENTENCES AND WORD

B.1. Thai sentence list

The following sentences are in a list which was read by each of five female native Thai speakers. The list is exactly as it appeared to the speakers when they recorded their readings.

Item #	Sentence
1	แดงอยากกินหมูปิ้ง
2	บ้านหลังนี้มีห้องนอนสามห้อง
3	ในทุ่งนามีควายสามตัว
4	ดำวิ่งเข้าไปกอดพ่อ
5	การถ่ายภาพต้องมีแสงดี
6	น้อยไม่ชอบที่มีหนุ่มมาตามตื๊อ
7	ดาโทรไปจองโต๊ะกินข้าวมื้อเย็น
8	ต้อยกินข้าวหลายชามจนท้องโต
9	คณะท่องเที่ยวมาช้าเพราะมัวไปถ่ายรูปแพะ
10	หน่องชอบทานข้าวราดพริก
11	ต้อยขับรถเร็วเลยต้องเสียค่าปรับราคาแพง
12	หน่อยชอบทานข้าวต้มใส่ตับหมู
13	แดงรู้สึกว่าหายใจไม่ออกเหมือนมีคอถูกใครบีบ
14	การออกแบบเสื้อผ้าเป็นเรื่องยาก
15	ครูเริญทำการไหว้ครูก่อนร่ายรำดาบ
16	เอวิ่งไล่ปูลมบนหาดทราย
17	ก้องปาก้อนหินแก้เบื่อ
18	ผมต้องการซื้อปืน

19 แดงรู้สึกเจ็บตาข้างซ้าย

20 การปลูกต้นไม้เป็นการช่วยเหลือแผ่นดิน

21 ชาวนาช่วยกันถอนต้นหญ้า

22 ชาวสวนช่วยกันตักน้ำใส่ถังเอาไปรดน้ำต้นไม้

23 ก้อนช่วยพ่อตักปุ๋ยใส่ถุง

24 คนไทยมักชอบเชื่อเรื่องผี

25 ภารโรงหาถังผงไม่เจอ

B.2. Thai sentence glosses

The following are the word-for-word glosses and translations of the Thai sentences. Each sentence is given in four lines. The first line is the Thai script, then the transcription[29], then a word to word gloss, and finally a translation of the sentences. In each sentence, the transcription that is the continuous stimulus is bolded.

1.

แดง	อยาก	กิน	หมู	ปิ้ง
Deng	yaak	kin	muu	**bping.F**
Deng	want	eat	pork	grilled

"Deng wants to eat grilled pork."

2.

บ้าน	หลัง	นี้	มี	ห้องนอน	สาม	ห้อง
baan.F	lang	nii	mii	haawng naawn	saam	haawng
house	CLF.building	this	have	bedroom	three	CLF.room

"This house has three bedrooms."

[29] The transcription method used for the Thai sentences can be found at http://www.thai-language.com/ref/transcribing-english.

3.

ใน	ทุ่ง	นา	มี	ควาย	สาม	ตัว
nai	**thoong.F**	naa	mii	kwai	saam	dtua
in	rice	field	have	buffalo	three	CLF.animal

"There are three buffalo in the rice field."

4.

ดำ	วิ่ง	เข้าไป	กอด	พ่อ
Dam	wing	khau bpai	kaawd	**phaaw.F**
Dam	run	into	hug	father

"Dam ran to hug father."

5.

การ	ถ่ายภาพ	ต้อง	มี	แสง	ดี
kaan	phai **phaap.F**	dtaawng	mii	saeng	dii
profession	photography	must	have	lighting	good

"Photography must have good lighting."

6.

น้อย	ไม่	ชอบ	ที่	มี	หนุ่ม	มา	ตามตื๊อ
Noi	mai	chaawp	thii	mii	num	maa	dtaam **dteuu.H**
Noi	NEG	like	that	have	young men	come	watch

"Noi does not like to have young men watching her."

7.

ดา	โทรไป	จอง	โต๊ะ	กิน	ข้าว	มื้อเย็น
Daa	thow bpai	jaawng	**dto.H**	kin	khau	mua yen
Daa	telephone	reserve	table	eat	food	dinner time

"Daa phoned to reserve a dinner table for dinner time."

8.

ต้อย	กิน	ข้าว	หลาย	ชาม	จน	ท้อง	โต
Dtoi	kin	khau	lai	chaam	jon	**thaawng.H**	dtoo
Doi	eat	food	many	plates	until	belly	full

"Doi ate many plates of food until she was full"

9.

คณะ	ท่องเที่ยว	มา	ช้า	เพราะ	มัวไป	ถ่ายรูป	แพะ
kɔna	thaawng thieo	maa	chaa	phraw	muabpai	thay ruub	**phae.H**
group	tourist	come	late	because	occupied	take picture	goat

"The tourist group came late because they were preoccupied with taking a picture of a goat."

10.

หน่อง	ชอบ	ทาน	ข้าว	ราด	พริก
Nong	chaawp	thaan	khau	raad	**phrik.H**
Nong	like	eat	rice	with	pepper

"Nong likes to eat pepper stir fry."

11.

ต้อย	ขับ	รถ	เร็ว	เลย	ต้อง	เสีย	ค่าปรับ	ราคา	แพง
Dtoi	khap	rot	reow	ley	dtaawng	siey	khaa **bprap.L**	raa khaa	phaeng
Doi	drive	car	fast	really	must	pay	fine	price	expensive

"Doi drove the car really fast and must pay an expensive fine."

12.

หน่อย	ชอบ	ทาน	ข้าวต้ม	ใส่	ตับ	หมู
Noi	chaawp	thaan	khau dtom	sai	**dtap.L**	muu
Noi	likes	eat	rice porridge	with	liver	pork

"Noi likes to eat rice porridge with pork liver."

13.

แดง	รู้สึก	ว่า	หายใจ	ไม่	ออก	เหมือน	มี	คอ	ถูก	ใคร	บีบ
Deng	ruuseuuk	waa	haaijaai	mai	awk	muan	mii	khaw	thuuk	krai	**beep.L**
Deng	feels	that	breath	neg	out	as if	have	neck	cause	someone	choke

"Deng felt that he could not breath as if someone was choking him"

14.

การ	ออก	แบบ	เสื้อผ้า	เป็น	เรื่อง	ยาก
kaan	awkbaeb	**baaep.L**	suaphaa	ben	ruang	yaak
thing	design	plan	clothes	is	item	difficult

"Designing clothing is a difficult thing."

15.

ครู	เริญ	ทำ	การ	ไหว้	ครู	ก่อน	ร่ายรำ	ดาบ
khru	Ruen	thaam	kaan	wai	khru	kawn	raairaam	**dtap.L**
Teacher	Ruen	makes	the	pay respect	teacher	before	dance	sword

"Teacher Ruen paid respect before the sword dancing."

16.

เอ	วิ่ง	ไล่	ปู	ลม	บน	หาด	ทราย
Aey	wing	lai	**bpuu.M**	lowm	bowb	haad	sai
Aey	run	chase	crab	wind	on	beach	sand

"Aey chased the crab on the windy beach."

17.

ก้อง	ปา	ก้อน	หิน	แก้	เบื่อ
Kong	**bpaa.M**	kawn	hin	kae	buua
Kong	throws	lump	stone	fix	boredome

"Kong throws rocks to reduce his boredome."

18.

ผม	ต้องการ	ซื้อ	ปืน
phowm	dtawngkaan	seuu	**bpeuun.M**
I	need	purchase	gun

"I need to purchase a gun."

19.

แดง	รู้สึก	เจ็บ	ตา	ข้าง	ซ้าย
Daeng	ruuseuuk	jep	**dtaa.M**	khaang	saai
Daeng	think	hurt	eye	side	left

"Daeng thinks he hurt his left eye."

20.

การ	ปลูก	ต้น	ไม้	เป็น	การ	ช่วย	เหลือ	แผ่น	ดิน
kaan	blook	dtown	mai	ben	kaan	chuay	luua	phaaen	**din.M**
thing	grow	plant	flower	be	thing	help		earth	

"Growing plants help the earth."

21.

ชาว	นา	ช่วย	กัน	ถอน	ต้น	หญ้า
chaow	naa	chuay	kan	**thaawn.R**	dtown	naa
farmer	field	help	each	pull	plant	weed

"Farmers help each other pull weeds."

22.

ชาว	สวน	ช่วย	กัน	ตัก	น้ำ	ใส่	ถัง	เอา	ไป	รด	น้ำ	ต้น	ไม้
chaow	suan	chuay	kan	dtak	nam	sai	**thang.R**	auw	bai	rot	nam	dtown	mai
farmer	animal	help	each	scoop	water	into	bucket	to	go	pour	water	plant	flower

"Farmers and animals help each other scoop water into buckets to pour water on the plants."

23.

ก้อน	ช่วย	พ่อ	ตัก	ปุ๋ย	ใส่	ถุง
Kon	chuay	phaaw	dtak	bpuy	sai	**thoong.R**
Kon	helps	father	scoop	fertilizer	into	bag

"Kon helps father scoop fertilizer into the bag."

24.

คน	ไทย	มัก	ชอบ	เชื่อ	เรื่อง	ผี
khon	thai	mak	chop	chua	rueng	**phee.R**
Person	Thai	tend to	like	believe	story	ghost

"Thai people like to believe ghost stories."

25.

ภาร	โรง	หา	ถัง	ผง	ไม่	เจอ
phaan	rowng	haa	thang	**phohng.R**	mai	jer
Janitor		seek	bucket	dust	cannot	find

"The janitor cannot find the dust bucket."

B.3. Thai Word List and Translations

Item #	Transcription	Orthography	Tone	Lexical Category	Definition
1	bprak	ปรัก	L	adjective	silver
2	bprap	ปรับ	L	verb	to adjust
3	bpraa	ปร่า	L	adjective	unpalatable
4	bpraak	ปราก	L	proper noun	Prague
5	bpraang	ปราง	M	noun	cheek
6	bpraang[k]	ปรางค์	M	noun	pagoda
7	bpraat[n]	ปราชญ์	L	noun	philosopher
8	bpraan	ปราณ	M	noun, loanword, Pali	inspiration
9	bpraap	ปราบ	L	verb	to suppress
10	bpraat	ปราศ	L	modifier	[is] deprived of
11	bpri	ปริ	L	verb	to open slightly
12	bprin[d]	ปริ๊นต์	H	loanword, English	to print
13	bprim	ปริ่ม	L	adjective	flush with
14	bpree	ปรี่	L	verb	to dash
15	bproo	ปรุ	L	adjective	perforated
16	bproong	ปรุง	M	verb	to season
17	bpruup	ปรู๊ฟ	H	loanword, English	proof
18	bplohng	ปลง	M	verb	to put an item down
19	bploht	ปลด	L	verb	to release
20	bplohn	ปล้น	F	verb	to rob
21	bplaaw	ปลอ	M	noun	Suay language
22	bplaawk	ปลอก	L	noun	collar
23	bplaawng	ปล่อง	L	noun	vertical shaft
24	bplaawng	ปล้อง	F	noun	a section of bamboo
25	bplaawt	ปลอด	L	adjective	lacking
26	bplaawp	ปลอบ	L	verb	to subdue
27	bplaawm	ปลอม	M	adjective	fake
28	bplak	ปลัก	L	noun	a muddy place
29	bplak	ปลั๊ก	H	noun, loanword, English	plug

30	bpling	ปลิง	M	noun	leech
31	bplit	ปลิด	L	verb	to pluck
32	bplin	ปลิ้น	F	verb	to turn inside-out
33	bplee	ปลี	M	noun	flower cluster
34	bplee	ปลี้	F		
35	bpleek	ปลีก	L	verb	to slip away
36	bpleuum	ปลื้ม	F	modifier	delighted
37	bplook	ปลุก	L	verb	to rouse
38	bpluuk	ปลูก	L	verb	to cultivate
39	bpaaw	ปอ	M		
40	bpoht	ปด	L	verb	to lie
41	bpohn	ปน	M	verb	to mix
42	bpohn	ป่น	L	verb	to pulverize
43	bpohp	ปบ	L	verb	to cover over
44	bpohm	ปม	M	noun	knot
45	bpaawk	ปอก	L	verb	to peel
46	bpaawng	ปอง	M	verb	to desire
47	bpaawng	ป่อง	L	verb	to bulge
48	bpaawng	ป้อง	F	verb	to protect
49	bpaawt	ปอด	L	noun	lungs
50	bpaawn	ป้อน	F	verb	to feed an animal
51	bpaawn[d]	ปอนด์	M	noun, loanword, English	pound
52	bpaawp	ปอบ	L	noun	an ogre
53	bpaawm	ป้อม	F	noun	fort
54	bpaaw[r]d	ปอร์ต	L	noun, loanword, English	port
55	bpa	ปะ	L	verb	mend
56	bpak	ปัก	L	verb	to embroider
57	bpak[s]	ปักษ์	L	noun	quarter of a moon
58	bpang	ปัง	M	noun	bread
59	bpat	ปัด	L		
60	bpat[m]	ปัทม์	L	noun, loanword, Pali	face
61	bpan	ปัน	M	verb	to ration

62	bpan	ปั่น	L	verb	to spin
63	bpan	ปั้น	F	verb	to mold
64	bpap	ปั๊บ	H	adverb	immediately
65	bpaa	ปา	M	verb	to throw
66	bpaa	ป่า	L	noun	forest
67	bpaa	ป้า	F	noun	aunt
68	bpaa	ป๊า	H	noun, colloquial	dad
69	bpaa	ป๋า	R	noun, loanword, Chinese	father
70	bpaak	ปาก	L	noun	mouth
71	bpaa[l]m	ปาล์ม	M	noun, loanword, English	palm
72	bpaat	ปาส	L		
73	bping	ปิ้ง	F	verb	to grill
74	bping	ปิ๊ง	H	verb	to like
75	bpit	ปิด	L	verb	to close
76	bpin	ปิ่น	L	noun	hair pin
77	bpee	ปี	M	noun	year
78	bpee	ปี่	L	noun	flute
79	bpee	ปี้	F	noun	gambling counter
80	bpeek	ปีก	L	noun	wing
81	bpeen	ปีน	M	verb	to climb
82	bpeep	ปีบ	L		
83	bpeep	ปี๊บ	H	noun	bucket
84	bpeuk	ปึก	L	noun	cake
85	bpeuk	ปึ้ก	F	adjective	closeness
86	bpeuut	ปีติ	L	noun	gladness
87	bpeuun	ปืน	M	noun	gun
88	bpeuun	ปื้น	F	classifier	handsaws
89	bpook	ปุก	L	noun	
90	bpook	ปุ๊ก	H	adjective	pudgy
91	bpoong	ปุ้ง	F	noun	gunshot
92	bpoop	ปุ๊บ	H	adverb	suddenly
93	bpoom	ปุ่ม	L	noun	knob

94	bpoom	ปุ้ม	F	modifier	round
95	bpuu	ปู	M	noun	crab
96	bpuu	ปู่	L	noun	grandfather
97	bpuut	ปูด	L	verb	to swell
98	bpuun	ปูน	M	noun	lime
99	bpeh	เป้	F	noun	backpack
100	bpeh	เป๋	R	adjective	twisted
101	bpehk	เป๊ก	H	noun	thumbtack
102	bpaehng	เปง	M		
103	bpehng	เป้ง	F	adjective	large
104	bpet	เป็ด	L	noun	duck
105	bpen	เป็น	M	verb	to be
106	bprohk	ปรก	L	verb	to cover
107	bprohng	ปรง	M	noun	the sago palm
108	bprohn	ปรน	M		
109	bprohp	ปรบ	L	verb	to clap
110	bpra	ประ	L	verb	sprinkle
111	bpraeht	เปรต	L	noun	ghost
112	bpraehm	เปรม	M	proper noun	Prem
113	bpruuhr	เปรอ	M	verb	to pander
114	bpruh	เปรอะ	L	adjective	soiled
115	bpraw	เปราะ	L	adjective	brittle
116	bplaeh	เปล	M	noun	cot
117	bplehng	เปล่ง	L	verb	to glow
118	bplaw	เปลาะ	L	noun	section
119	bpeh	เป๊ะ	H	adverb	precisely
120	bpaw	เปาะ	L	adverb	praise
121	bpeerk	เปิก	L	adjective	abraded
122	bpeert	เปิด	L	verb	open
123	bpeern	เปิ่น	L	adjective	embarrassed
124	bpeerp	เปิบ	L	verb	eat
125	bpeern	เปี้ล	F	noun, loanword, English	

126	bpaae	แป	M	noun	purlin
127	bpaaeng	แป้ง	F	noun	powder
128	bpaaet	๘	L	number	eight
129	bpaaet	แป๊ด	H	noun	horn blast
130	bpaaen	แป้น	F	noun	washer
131	bpaaep	แป๊บ	H	adverb	momentarily
132	bpraae	แปร	M	verb	amend
133	bpraaeng	แปรง	M	verb	brush
134	bpraeng	แปร่ง	L	modifier	discordant
135	bplaae	แปล	M	verb	translate
136	bplaaek	แปลก	L	modifier	weird
137	bplaaeng	แปลง	M	noun	field
138	bplaaen	แปลน	M	noun, loanword, English	
139	bpae	แป๊ะ	H	noun, loanword, Chinese	uncle
140	bpoxh	โป๊	H	adjective	naked
141	bpoxhng	โป่ง	L	verb	inflate
142	bpoxhng	โป้ง	F	noun	thumb
143	bpoxhp	โป๊ป	H	noun, loanword, English	the Pope
144	bproxhng	โปร่ง	L	modifier	spacious
145	bproxht	โปรด	L	verb	favor
146	bpo	โป๊ะ	H	noun	lampshade
147	bpohk	ปก	L	noun	book covering
148	bpohng	ปง	M		
149	dtaaw	ฏอ	M		
150	dtwat	ตวัด	L		
151	dtaaw	ต่อ	L	preposition	against
152	dtaaw	ต้อ	F	noun	cataract
153	dtaawk	ตอก	L	noun	strip of bamboo
154	dtaawk	ต๊อก	H	noun	guinea pig
155	dtaawng	ตอง	M	noun	a trick of three cards
156	dtaawng	ต้อง	F	verb	must
157	dtaawng	ต๊อง	H	noun	semen

158	dtaawt	ตอด	L	verb	nibble
159	dtaawn	ตอน	M	noun	interval
160	dtaawn	ต้อน	F	verb	herd
161	dtaawp	ตอบ	L	verb	reply
162	dtaawm	ตอม	M	verb	swarm around
163	dtaawm	ต่อม	L	noun	gland
164	dtaawm	ต้อม	F	proper noun	
165	dta	ตะ	L	verb	encrust
166	dtak	ตัก	L	verb	scoop
167	dtang	ตัง	M	noun	mat
168	dtang	ตั่ง	L	noun	small table
169	dtang	ตั้ง	F	verb	decree
170	dtang[k]	ตังค์	M	noun	a Thai coin
171	dtat	ตัด	L	verb	cut
172	dtan	ตัน	M	modifier	clogged
173	dtap	ตับ	L	noun	liver
174	dtoht	ตด	L	verb	fart
175	dtohn	ตน	M	noun	oneself
176	dtohn	ต้น	F	noun	plant
177	dtaa	ตา	M	noun	eye
178	dtaak	ตาก	L	verb	expose
179	dtaang	ต่าง	L	adverb	various
180	dtaan	ตาน	M	noun	disease
181	dtaan	ต้าน	F	verb	withstand
182	dtaap	ตาบ	L	noun	embroidered emblem
183	dti	ติ	L	verb	criticize
184	dting	ติง	M	verb	cross-examine
185	dting	ติ่ง	L	noun	protrusion
186	dtit	ติด	L	verb	be obsessed
187	dtim	ติ๋ม	R	proper noun	Tim
188	dtee	ตี	M	verb	beat
189	dtee	ตี่	L	noun	a running game

190	dtee	ตี๋	R	noun, loanword, Chinese	
191	dteen	ตีน	M	noun	foot
192	dteep	ตีบ	L	adjective	constricted
193	dteuk	ตึก	L	noun	building
194	dteung	ตึง	M	modifier	tight
195	dteuut	ตืด	L	noun	tapeworm
196	dteuun	ตื่น	L	verb	awaken
197	dteuun	ตื้น	F	noun	shallow
198	dteuu	ตื้อ	F	noun	stupid
199	dteuu	ตื๊อ	H	verb	harass
200	dteuu	ตื๋อ	R	adverb	pitch black
201	dtoo	ตุ	L	adjective	musky
202	dtoo	ตุ๊	H	adverb	chubby
203	dtook	ตุ๊ก	H	noun	tuk-tuk
204	dtoong	ตุง	M	adjective	bulging
205	dtoon	ตุน	M	verb	stockpile
206	dtoon	ตุ่น	L	noun	bamboo rat
207	dtoon	ตุ๋น	R	verb	stew
208	dtoom	ตุ่ม	L	noun	carp
209	dtoom	ตุ้ม	F	noun	pendant
210	dtuu	ตู่	L	verb	usurp
211	dtuu	ตู้	F	noun, loanword, Chinese	cupboard
212	dtuut	ตูด	L	noun	buttocks
213	dtuup	ตูบ	L	verb	droop down
214	dtuum	ตูม	M	adjective	young
215	dteerp	เติบ	L	adjective	lavish
216	dteerm	เติม	M	verb	fill up
217	dtaae	แต่	L	conjunction	but
218	dtaaek	แตก	L	verb	smash
219	dtaaeng	แตง	M	noun	squash
220	dtaeng	แต่ง	L	verb	decorate
221	dtaaen	แตน	M	noun	wasp

222	dtaaem	แต้ม	F	noun	point
223	dtraae	แตร	M	noun	horn
224	dtae	แตะ	L	verb	touch
225	dtoxh	โต	M	modifier	large
226	dtoxh	โต้	F	verb	argue
227	dtoxhng	โต่ง	L	adjective	extreme
228	dtoxhng	โต้ง	F	adjective	tall
229	dto	โต๊ะ	H	noun, loanword, Chinese	table
230	dtohp	ตบ	L	verb	slap
231	dtohm	ตม	M	noun	swamp
232	dtohm	ต้ม	F	verb	boil
233	dtrohng	ตรง	M	adjective	straight
234	dtrohm	ตรม	M	verb	mourn
235	dtraawk	ตรอก	L	noun	alley
236	dtraawng	ตรอง	M	verb	consider
237	dtrang	ตรัง	M	proper noun	Trang
238	dtrap	ตรับ	L	verb	listen to
239	dtrat	ตรัส	L	verb	remark
240	dtraa	ตรา	M	noun	insignia
241	dtraat	ตราด	L	proper noun	Trat
242	dtraap	ตราบ	L	preposition	until
243	dtree	ตรี	M	adjective, loanword, Pali	three
244	dtreuk	ตรึก	L	verb	consider
245	dtreung	ตรึง	M	verb	bind
246	dtreum	ตรึม	M	adverb	abundantly
247	dtroo	ตรุ	L	noun	dungeon
248	dtroot	ตรุษ	L		
249	dtruu	ตรู	M	adjective	beautiful
250	dtruu	ตรู่	L	adjective	very early
251	dtohk	ตก	L	verb	fall
252	dtohng	ตง	M	noun	girder
253	dtohng	ต๋ง	R	verb, loanword, Chinese	collect a fee from gambling winnings

254	baat[r]	บาตร	L	noun, loanword, Pali	the bowl carried by a Buddhist monk
255	baan	บาน	M	verb	bloom
256	baan	บ้าน	F	noun	home
257	baap	บาป	L	noun	sin
258	baa[r]	บาร์	M	noun, loanword, English	bar
259	baa[r]f	บาร์ฟ	L	noun, loanword, English	barf
260	baat[k]	บาศก์	L	noun	dice
261	bi	บิ	L	verb	split
262	bik	บิ๊ก	H	proper noun	Bik
263	bit	บิฐ	L	noun	stool
264	bin	บิน	M	verb	fly
265	bin	บิ่น	L	modifier	chipped
266	bin[l]	บิลล์	M		
267	bee	บี	M		B
268	bee	บี้	F	verb	crush
269	beep	บีบ	L	verb	squeeze
270	beuk	บึก	L	noun	Mekong giant catfish
271	beung	บึง	M	noun	swamp
272	beung	บึ้ง	F	adjective	serious
273	beum	บึ้ม	F	onomatopoeia	bang
274	beuu	บื้อ	F	adjective	stupid
275	book	บุก	L	verb	invade
276	book	บุ๊ค	H	verb	reserve
277	boong	บุ้ง	F	noun	hairy caterpillar
278	boon	บุญ	M	noun	merit
279	boot[r]	บุตร	L	noun, loanword, Pali	offspring
280	boon	บุ๋น	R	adjective	
281	boop	บุบ	L	verb	grind
282	boom	บุม	M	loanword, English	boom
283	boom	บุ๋ม	R	modifier	pock-marked
284	braehk	เบรก	L		

285	blaehs	เบลซ	L	noun, loanword, English	blaze
286	baeht	เบส	L	noun, loanword, English	base
287	buuhr	เบ้อ	F	adjective	gaping
288	buuhr	เบอร์	M	noun, loanword, English	number
289	buh	เบอะ	L	adjective	huge
290	buh	เบ๊อะ	H	adjective	stupid-looking
291	beh	เบะ	L	adjective	excess
292	baw	เบาะ	L	noun	cushion
293	beerk	เบิก	L	verb	widen
294	beerng	เบิ่ง	L	verb	gaze
295	beert	เบิช	L		
296	beerm	เบิ้ม	F	adjective	huge
297	beern	เบิร์น	M	proper noun	Bern
298	baae	แบ	M	verb	display
299	baaek	แบก	L	verb	carry piggyback
300	baeng	แบ่ง	L	verb	share
301	baaeng[k]	แบงค์	M		
302	baaen	แบน	M	modifier	flat
303	baaep	แบบ	L	noun	type
304	bohm	บ่ม	L	verb	cure fruits
305	breef	บรีฟ	L	noun, loanword, English	brief
306	bruus	บรู๊ซ	H		
307	blawk	บล็อค	L	noun, loanword, English	block
308	baaw	บอ	M		
309	baaw	บ่อ	L	noun	pond
310	baaw	บ้อ	F	modifier	impoverished
311	baaw	บ๊อ	H	particle	question
312	baawk	บอก	L	verb	tell
313	baawng	บ้อง	F	noun	marijuana pipe
314	baawng	บ๊อง	H	adjective	crazy
315	baawt	บอด	L	adjective	weakened

316	baawn	บอน	M	noun	Caladium plant
317	baawn	บ่อน	L	noun	casino
318	baawn[n]	บอนน์	M	proper noun	Bonn
319	baawp	บอบ	L	modifier	badly injured
320	baawp	บ๊อบ	H	proper noun	Bob
321	baawm[p]	บอมบ์	M	noun, loanword, English	bomb
322	baaw[r]d	บอร์ด	L	noun, loanword, English	board
323	baawn[l]	บอลล์	M		
324	ba	บ๊ะ	H	interjection	darn!
325	bak	บัก	L	noun	derogatory honorific
326	bang	บัง	M	verb	conceal
327	bang	บั้ง	F	noun	rank insignia
328	baa	บ่า	L	noun	shoulder
329	baa	บ้า	F	modifier	crazy
330	baak	บาก	L	verb	mark
331	baang	บาง	M	adjective	few
332	baang	บ่าง	L	noun	flying squirrel
333	baang	บ้าง	F	adjective	some
334	bohk	บก	L	noun	dry land
335	bohng	บ่ง	L	verb	puncture
336	boht	บด	L	noun	small boat
337	bohn	บน	M	preposition	on
338	bohn	บ่น	L	verb	complain
339	daaw	ฎอ	M		
340	doon	ดุน	M	modifier	embossed
341	doon	ดุ้น	F	noun	kindling
342	doom	ดุม	M	noun	hub
343	doom	ดุ่ม	L	adverb	intently
344	doon[y]	ดุลย์	M		
345	duu	ดู	M	verb	look at
346	duut	ดูด	L	verb	suck
347	dek	เด็ก	L	noun	child

348	dehng	เด้ง	F	verb	bounce
349	det	เด็จ	L	verb	snap off
350	daeht	เดช	L	noun	power
351	deht	เด๊ด	H	adjective, loanword, English	dead
352	daehn	เดน	M	noun	leftovers
353	dehn	เด่น	L	modifier	prominent
354	daehf	เดฟ	L	proper noun	Dave
355	daehn[l]	เดลล์	M	noun, loanword, English	dell
356	duuhr	เด้อ	F	particle	polite particle
357	duuhr	เด้อ	R	modifier	clumsy
358	deh	เด่ะ	L		
359	daw	เดาะ	L	verb	toss
360	deern	เดิน	M	verb	walk
361	deern	เดิ้น	F	adjective, loanword, English	modern
362	deerm	เดิม	M	noun, loanword, Khmer	beginning
363	daae	แด่	L	preposition	for
364	daaek	แดก	L	verb	devour
365	daaeng	แดง	M	adjective	red
366	daaet	แดด	L	noun	sunshine
367	daaen	แดน	M	noun	region
368	daen	แด่น	L	verb	reach
369	dae	แดะ	L	verb	move side to side
370	doxh	โด	M	noun, loanword, Latin	do
371	doxh	โด่	L	adjective	erect
372	doxhng	โด่ง	L	verb	rise up
373	doxht	โดด	L	adjective	single
374	doxhn	โดน	M	verb	collide
375	do	โด๊ะ	H	verb	
376	daawk	ดอก	L	noun	flower
377	dawk	ด็อก	L	noun, loanword, English	dog
378	daawng	ดอง	M	verb	pickle

379	daawt	ดอด	L	verb	sneak into
380	dawt	ด็อต	L	noun, loanword, English	dot
381	daawn	ดอน	M	noun	mound
382	daawm	ดอม	M	noun, loanword, Cambodian	clothesline
383	daawm	ด้อม	F	verb	snoop
384	daawn[l]	ดอลล์	M	noun, loanword, English	dollar
385	daa	ดา	M	verb	advance in a group
386	daa	ด่า	L	verb	scold
387	daa	ด๋า	R		
388	daang	ด่าง	L	noun	lye
389	daat	ดาด	L	verb	pave
390	daan	ดาน	M	noun	wedge-shaped bolt
391	daan	ด่าน	L	noun	checkpoint
392	daan	ด้าน	F	modifier	hard-hearted
393	daap	ดาบ	L	noun	sword
394	di	ดิ	L	particle	request
395	dik	ดิก	L	adverb	closely
396	dik	ดิ๊ก	H	proper noun	Dick
397	ding	ดิ่ง	L	noun	plumb bob
398	din	ดิน	M	noun	dirt
399	din	ดิ้น	F	verb	struggle
400	deen	ดีน	M	proper noun	Dean
401	deuk	ดึก	L	adverb	dark
402	deung	ดึง	M	verb	pull
403	deung	ดิ่ง	L	verb	fall directly down
404	deum	ดื่ม	L	modifier	deep
405	deuun	ดื่น	L	modifier	abundant
406	deuum	ดื่ม	L	verb	drink
407	deuu	ดื้อ	F	modifier	stubborn
408	doo	ดุ	L	adjective	fierce
409	dook	ดุก	L	noun	catfish
410	doong	ดุ้ง	F	modifier	arched outwards

411	doot	ดุจ	L	preposition	as if
412	dohk	ดก	L	modifier	fertile
413	dohng	ดง	M	noun	jungle
414	dohn	ด้น	F	verb	push forward
415	dohm	ดม	M	verb	sniff
416	dohn	ดล	M	noun, loanword, Pali	floor
417	thaaw	ฐอ	R		
418	thaan	ฐาน	R	noun, loanword, Pali	pedestal
419	thaaw	ฑอ	M		
420	thoxh	โถ	R		
421	thoxhng	โถง	R	adjective	spacious
422	thoxhp	โถบ	L	verb	swoop down onto
423	thoxhm	โถม	R	verb	swoop down onto
424	thohm	ถ่ม	L	verb	expectorate
425	thaaw	ถ่อ	L	verb	pole a barge
426	thaawk	ถอก	L	verb	peel
427	thaawt	ถอด	L	verb	take off
428	thaawn	ถอน	R	verb	withdraw
429	thaawm	ถ่อม	L	verb	humble
430	tha	ถะ	L	noun, loanword, Chinese	pagoda
431	thak	ถัก	L	verb	weave
432	thang	ถัง	R	noun	bucket
433	that	ถัด	L	adjective	next
434	than	ถัน	R	noun	breast milk
435	than	ถั่น	L	adverb	rapidly
436	thap	ถับ	L	adjective	immediate
437	thaa	ถ้า	F	conjunction	conditional
438	thaang	ถาง	R	verb	cut grass
439	thaang	ถ่าง	L	verb	widen
440	thaat	ถาด	L	noun	tray
441	thaan	ถ่าน	L	noun	charcoal
442	thin	ถิ่น	L	noun	location

443	thee	ถี่	L	adjective	frequent
444	theep	ถีบ	L	verb	push off with the foot
445	theuk	ถึก	L	noun	water buffalo
446	theung	ถึง	R	verb	arrive
447	theuu	ถือ	R	verb	carry
448	thoong	ถุง	R	noun	bag
449	thoon	ถุน	R	verb	consume opium
450	thuu	ถู	R	verb	scrub
451	thuuk	ถูก	L	modifier	correct
452	thaehn	เถน	R	noun	thief
453	thuh	เถอะ	L	particle	a command
454	thaw	เถาะ	L	noun	rabbit
455	theerk	เถิก	L		
456	theert	เถิด	L	particle	a command
457	theern	เถิน	R	adjective	high
458	thaaep	แถบ	L	noun	collar
459	thaaem	แถม	R	conjunction	as well as
460	thohk	ถก	L	verb	dispute
461	thoht	ถด	L	verb	budge
462	thohm	ถม	R		
463	thap	ทับ	H	verb	superimpose
464	that[n]	ทัศน์	H	noun, loanword, Sanskrit	vision
465	thaa	ทา	M	verb	paint
466	thaa	ท้า	H	verb	dare
467	thaak	ทาก	F	noun	leech
468	thaang	ทาง	M	noun	way
469	thaan	ทาน	M	noun	alms
470	than	ท่าน	F	pronoun	3 person singular
471	thaap	ทาบ	F	verb	brace
472	thaa[r]	ทาร์	M	noun, loanword, English	tar
473	thaat	ทาส	F	noun	slave
474	thing	ทิ้ง	H	verb	discard

475	thit	ทิด	H	noun	secular life
476	thin	ทิน	M	noun, loanword, Pali	day
477	thip	ทิป	H	verb	tread]
478	thip[y]	ทิพย์	H	adjective	spiritual
479	thim	ทิม	M	proper noun	Tim
480	thim	ทิ่ม	F	verb	stab
481	thee	ที	M	noun	opportunity
482	thee	ที่	F	preposition	at
483	theem	ทีม	M	noun, loanword, English	team
484	theuk	ทึก	H	verb	state falsely
485	theung	ทึ่ง	F	modifier	enjoying
486	theung	ทึ้ง	H	verb	pull
487	theup	ทึบ	H	adjective	thick
488	theum	ทึม	M	adjective	dull
489	theum	ทึ่ม	F	adjective	stupid
490	theuu	ทื่อ	F	modifier	dull
491	thoo	ทุ	H	adjective, loanword, Pali	ill
492	thook	ทุก	H	pronoun	each
493	thook[k]	ทุกข์	H	noun	suffering
494	thoong	ทุ่ง	F	noun	field
495	thoot	ทุด	H	interjection	express contempt
496	thoon	ทุน	M	noun	assets
497	thoon	ทุ่น	F	noun	buoy
498	thoop	ทุบ	H	verb	pound
499	thoom	ทุม	M	noun, loanword, Pali	tree
500	thoom	ทุ่ม	F	noun	evening hours
501	thoom	ทุ้ม	H	adjective	low-frequency sound
502	thuu	ทู	M	noun	mackerel
503	thuu	ทู่	F	adjective	blunt
504	thuut	ทูต	F	noun	ambassador
505	thuun	ทูน	M	verb	respect
506	thuum	ทูม	M	adjective	swollen

507	thaeh	เท	M	verb	pour
508	theh	เท่	F	adjective	hip
509	thaehk	เทค	F	verb, loanword, English	take
510	thet	เท็จ	H	noun	untrue
511	thaehp	เทป	F	noun, loanword, English	audio cassette tape
512	thaehm[s]	เทมส์	M		
513	thraeht	เทรด	F	verb, loanword, English	trade
514	theerng	เทิง	M	adjective	large
515	theerng	เทิ่ง	F		
516	theert	เถิด	F	verb	honor
517	theern	เทิน	M	noun	mound
518	thaae	แท้	H	adjective	genuine
519	thaaeng	แทง	M	verb	stab
520	thaaeng	แท่ง	F	modifier	solid
521	thaaeng	แท้ง	H	verb	miscarriage
522	thaeng[k]	แท็งค์	M	noun, loanword, English	tank
523	thaaen	แทน	M	adjective	substitute
524	thaaen	แท่น	F	noun	dais
525	thaaep	แทบ	F	adverb	almost
526	thaep	แท็บ	H	noun, loanword, English	tab
527	thraaek	แทรค	F		
528	thraek	แทร็ค	H	noun, loanword, English	track
529	thae	แทะ	H	verb	gnaw
530	thoxh	โท	M	noun	telephone number
531	thoxht	โทษ	F	noun, loanword, Pali	punishment
532	thoht	ทศ	H	adjective, loanword, Sanskrit	ten
533	thaaw	ท่อ	F	noun	ditch
534	thaaw	ท้อ	H	modifier	sad
535	thaawng	ทอง	M	noun	gold
536	thaawng	ท่อง	F	verb	wander

537	thaawng	ท้อง	H	noun	belly
538	thaawt	ทอด	F	modifier	deep-fried
539	thaawt	ท้อด	H	proper noun	Todd
540	thaawn	ทอน	M	noun	change
541	thaawn	ท่อน	F	noun	portion
542	thaawp	ท้อป	H	noun, loanword, English	top
543	thaawm	ทอม	M	proper noun	Tom
544	thak	ทัก	H	verb	greet
545	thang	ทั่ง	F	noun	anvil
546	thang	ทั้ง	H	adjective	entire
547	than[t]	ทัณฑ์	M	noun, loanword, Pali	crime
548	that	ทัด	H	verb	wear
549	than	ทัน	M	noun	catch up with
550	than[d]	ทันต์	M	noun, loanword, Pali	teeth
551	thohn	ทน	M	verb	tolerate
552	thohn	ท้น	H	adjective	flooded
553	thohn[d]	ทนต์	M	noun, loanword, Pali	teeth
554	thohp	ทบ	H	verb	double over
555	than[m]	ธรรม์	M		
556	than[w]	ธันว์	M	noun, loanword, Pali	Sagittarius
557	thuup	ธูป	F	noun	incense
558	thuuhr	เธอ	M	pronoun	2 person singular
559	thoxh	โธ่	F		Damn!
560	tha	ธ	H	pronoun	[3 person royalty
561	thohng	ธง	M	noun	flag
562	phin	ผิน	R	verb	twist
563	phee	ผี	R	noun	ghost
564	pheung	ผึ่ง	L	verb	sun-dry
565	pheung	ผึ้ง	F	noun	bee
566	pheuun	ผืน	R	noun	sheet
567	pheuun	ผื่น	L	noun	rash
568	phoo	ผุ	L	adjective	rotten

569	phoot	ผุด	L	verb	pop up
570	phuu	ผู้	F	noun	person
571	phuuk	ผูก	L	verb	bind
572	phaeh	เผ	R	noun	Thai card game
573	phaehng	เผง	R	adverb	completely
574	phet	เผ็ด	L	modifier	spicy-hot
575	phehn	เผ่น	L	verb	rush
576	phluuhr	ผลอ	R	modifier	reckless
577	pheern	เผิน	R	adjective	cursory
578	phaae	แผ่	L	verb	expand
579	phaaeng	แผง	R	noun	stall
580	phaaet	แผด	L	verb	roar
581	phaaen	แผน	R	noun	plan
582	phaen	แผ่น	L	noun	sheet
583	phlaae	แผล	R	noun	wound
584	phlaaeng	แผลง	R	verb	modify
585	phoxh	โผ	R	verb	leap
586	phoxhng	โผง	R	noun	bang
587	phoxhn	โผน	R	verb	fly
588	phloxh	โผล่	L	verb	appear
589	phla	ผละ	L	verb	leave
590	phlak	ผลัก	L	verb	push
591	phlat	ผลัด	L	verb	take turns
592	phlap	ผลับ	L	adjective	quick
593	phlaan	ผลาญ	R	verb	waste
594	phli	ผลิ	L	verb	sprout
595	phloong	ผลุง	R	adverb	suddenly
596	phloop	ผลุบ	L	verb	dive down
597	phaaw	ผอ	R		
598	phaawk	ผอก	L	noun	sheath
599	phaawng	ผอง	R	adverb	whole
600	phaawng	ผ่อง	L	modifier	bright
601	phaawn	ผ่อน	L	verb	make payments

602	phaawm	ผอม	R	modifier	skinny
603	phak	ผัก	L	noun	vegetable
604	phang	ผัง	R	noun	chart
605	phat	ผัด	L	verb	stir fry
606	phan	ผัน	R	verb	change
607	phap	ผับ	L	noun, loanword, English	pub
608	phaa	ผา	R	noun	cliff
609	phaa	ผ่า	L	verb	cleave
610	phaa	ผ้า	F	noun	clothing
611	phaak	ผาก	L	adjective	dry
612	phaang	ผาง	R	onomatopoeia	dull bang
613	phaat	ผาด	L	adjective	indirect
614	phaan	ผ่าน	L	verb	cross
615	phaan	ผาล	R	noun, Pali, Sanskrit	ploughshare
616	phing	ผิง	R	verb	bake
617	phit	ผิด	L	modifier	wrong
618	phohng	ผง	R	noun	dust
619	phoht	ผด	L	noun	blister
620	phohm	ผม	R	pronoun	1 person male
621	phlohp	พลบ	H	noun	dusk
622	phlaawng	พลอง	M	noun	staff
623	phlaawt	พลอด	F	verb	pleasant conversation
624	phlawm	พล็อม	M	prefix	
625	phlak	พลั่ก	F	adverb	generously
626	phlang	พลั่ง	F	adverb	copiously
627	phlang	พลั้ง	H	verb	err
628	phlat	พลัด	H	verb	trip
629	phlan	พลัน	M	adverb	unexpectedly
630	phlap	พลับ	H	noun	persimmon
631	phlaa	พล่า	F	noun	raw beef salad
632	phlaang	พลาง	M	adverb	simultaneously
633	phlaat	พลาด	F	verb	miss

634	phlaan	พล่าน	F	adverb	turbulently
635	phlik	พลิก	H	verb	twist
636	phlee	พลี	M	verb	sacrifice
637	phloo	พลุ	H	noun	fireworks
638	phloong	พลุ่ง	F	verb	gush
639	phluu	พลู	M	noun	Betel leaf
640	phaaw	พอ	M	modifier	enough
641	phaaw	พ่อ	F	noun	father
642	phaaw	พ้อ	H	verb	complain
643	phaawk	พอก	F	verb	cover over
644	phaawng	พอง	M	modifier	inflated
645	phaawng	พ้อง	H	modifier	identical
646	phaawt	พอด	F	noun, loanword, English	pod
647	phaawn	พอน	M		
648	phaawp	พ๊อพ	H	noun, loanword, English	pop
649	phak	พัก	H	verb	halt
650	phak[dr]	พักตร์	H	noun	face
651	phang	พัง	M	verb	crash
652	phat	พัด	H	verb	fan
653	phan	พัน	M	noun	thousand
654	phan[ch]	พันช์	M	noun, loanword, English	punch
655	phan[t]	พันธ์	M	verb	bind
656	phap	พับ	H	verb	fold
657	phat[dr]	พัสตร์	H	noun	cloth
658	phaa	พา	M	verb	lead the way
659	phaak	พาก	F	prefix	
660	phaak[y]	พากย์	F	noun	commentary
661	phaang	พ่าง	F	conjunction	as
662	phaat	พาด	F	verb	lean on
663	phaan	พาน	M	noun	tray
664	phing	พิง	M	verb	rest upon
665	phit[n]	พิชญ์	H	noun, loanword, Pali	savant

666	phin	พิณ	M	noun	guitar
667	phohp	พบ	H	verb	meet
668	phim[p]	พิมพ์	M	verb	print
669	phit	พิศ	H	verb	stare at
670	phee	พี	M		
671	phee	พี่	F	noun	older sibling
672	pheet	พีช	F	noun, loanword, English	peach
673	phees	พีซ	F	noun, loanword, English	piece
674	pheung	พึง	M	verb	ought to
675	pheum	พึม	M	onomatopoeia	murmuring
676	pheuut	พืช	F	noun, loanword, Pali	vegetation
677	pheuun	พื้น	H	noun	ground
678	pheuun	พื้น	M	noun	floor
679	phoo	พุ	H	verb	erupt
680	phook	พุก	H	noun	a cleat for tying a line
681	phoong	พุง	M	noun	paunch
682	phoong	พุ่ง	F	verb	throw
683	phoot	พุทธ	H	noun	Buddha
684	phoom	พุ่ม	F	noun	shrubbery
685	phuu	พู	M	noun	section
686	phuut	พูด	F	verb	speak
687	phuun	พูน	M	verb	pile up
688	phehng	เพ่ง	F	verb	gaze
689	phaeht	เพจ	F	noun, loanword, English	page
690	phet	เพ็จ	H	adjective	small
691	pheht	เพชร	H	noun	diamond
692	phen	เพ็ญ	M	adjective	full moon
693	phraw	เพราะ	H	conjunction	because
694	phaehn	เพล	M	noun	last meal of the day
695	phleh	เพล่	F	adjective	oblique
696	phlaehng	เพลง	M	noun	song
697	phlaeh[y]	เพลย์	M	noun, loanword, English	play

698	phlaw	เพลาะ	H	verb	splice together
699	phleerng	เพลิง	M	noun	fire
700	phleert	เพลิด	F	particle	
701	phleern	เพลิน	M	verb	enjoy
702	phuuhr	เพ่อ	F	adverb	until now
703	phuuhr	เพ้อ	H	verb	be delirious
704	pheh	เพะ	H	verb	throw
705	phaw	เพาะ	H	verb	cultivate
706	pheerk	เพิก	F	verb	revoke
707	pheerng	เพิง	M	noun	shed
708	pheerng	เพิ่ง	F	verb	recently
709	pheern	เพิ่น	F	pronoun	3 person singular
710	pheerm	เพิ่ม	F	verb	increase
711	phaae	แพ	M	noun	raft
712	phaae	แพ้	H	verb	allergic to
713	phaaeng	แพง	M	modifier	expensive
714	phaaeng	แพ่ง	F	adjective	civil
715	phaaet	แพท	F	proper noun	Pat
716	phaaet[y]	แพศย์	F	noun	medical doctor
717	phaaen	แพน	M	noun	sheet
718	phaaen	แพ่น	F	verb	strike
719	phraae	แพร	M	noun	silk
720	phaae[r]	แพร์	M	noun, loanword, English	pear
721	phraae	แพร่	F	proper noun	Phrae
722	phraaeng	แพร่ง	F	noun	cross-road
723	phlaaeng	แพลง	M	verb	sprain
724	phlaaem	แพลม	M	verb	protrude
725	phlaem	แพล็ม	M	modifier	protruding
726	phae	แพะ	H	noun	goat
727	phoxhk	โพก	F	verb	wrap a piece of cloth around the head
728	phoxhng	โพง	M	verb	scoop
729	phoxht	โพธิ์	F	noun	Bhodi tree

730	phoxhn	โพน	M	verb	catch
731	phoxhn	โพ้น	H	adverb	far away
732	phroxhk	โพรก	F	noun	hole
733	phroxhng	โพรง	M	noun	hole
734	phloxhng	โพลง	M	modifier	vivid
735	phloxhng	โพล่ง	F	onomatopoeia	plop!
736	phloxhn	โพลน	M	modifier	glowing
737	phoxht[d]	โพสต์	F	noun, loanword, English	post
738	phroht	พรต	H	noun, loanword, Sanskrit	wish
739	phrohm	พรม	M	noun	carpet
740	phan[k]	พรรค์	M	noun, loanword, Pali	kind
741	phraawng	พร่อง	F	verb	become empty
742	phraawng	พร้อง	H	verb	speak
743	phraawm	พร้อม	H	adjective	ready
744	phra	พระ	H	noun	monk
745	phran	พรั่น	F	modifier	scared
746	phraa	พร่า	F	verb	destroy
747	phraa	พร้า	H	noun	scythe
748	phraak	พราก	F	verb	separate
749	phraang	พราง	M	verb	hide
750	phraang	พร่าง	F	adjective	brilliant
751	phraan	พราน	M	noun	hunter
752	phrik	พริก	H	noun	chili pepper
753	phohk	พก	H	noun	pocket
754	phohng	พง	M	noun	forest undergrowth
755	phohng[s]	พงศ์	M	noun	history
756	phoht[n]	พจน์	H	noun, loanword, Pali	speech
757	phohn	พ่น	F	verb	blow
758	phohn	พ้น	H	verb	pass by
759	phat[r]	ภัตร	H	noun	food
760	phaap	ภาพ	F	noun	image
761	phruu	ภรู	M	noun	eyebrow

762	phak[s]	ภักษ์	H	verb	eat

APPENDIX C: DATABASES

C.1. Tokens

The following list represents the recorded tokens from all five speakers for the entire database. The website, http://www.thai-language.com/dict/, was the reference source for these tokens. The Romanization scheme for these tokens was developed at the website with documentation provided at http://www.thai-language.com/ref/phonemic-transcription.

C.1.1. Token database

1-bprak,L
2-bprap,L
3-bpraa,L
4-bpraak,L
5-bpraang,M
6-bpraang[k],M
7-bpraat[n],L
8-bpraan,M
9-bpraap,L
10-bpraat,L
11-bpri,L
12-bprin[d],H
13-bprim,L
14-bpree,L
15-bproo,L
16-bproong,M
17-bpruup,H
18-bplohng,M
19-bploht,L
20-bplohn,F
21-bplaaw,M
22-bplaawk,L
23-bplaawng,L
24-bplaawng,F
25-bplaawt,L
26-bplaawp,L
27-bplaawm,M
28-bplak,L
29-bplak,H
30-bpling,M
31-bplit,L
32-bplin,F
33-bplee,M
34-bplee,F
35-bpleek,L
36-bpleuum,F
37-bplook,L
38-bpluuk,L
39-bpaaw,M
40-bpoht,L
41-bpohn,M
42-bpohn,L
43-bpohp,L
44-bpohm,M
45-bpaawk,L
46-bpaawng,M
47-bpaawng,L
48-bpaawng,F
49-bpaawt,L
50-bpaawn,F
51-bpaawn[d],M
52-bpaawp,L
53-bpaawm,F
54-bpaaw[r]d,L
55-bpa,L
56-bpak,L
57-bpak[s],L
58-bpang,M
59-bpat,L
60-bpat[m],L
61-bpan,M
62-bpan,L
63-bpan,F
64-bpap,H
65-bpaa,M
66-bpaa,L
67-bpaa,F
68-bpaa,H
69-bpaa,R
70-bpaak,L
71-bpaa[l]m,M
72-bpaat,L
73-bping,F
74-bping,H
75-bpit,L
76-bpin,L
77-bpee,M
78-bpee,L
79-bpee,F
80-bpeek,L
81-bpeen,M
82-bpeep,L
83-bpeep,H
84-bpeuk,L
85-bpeuk,F
86-bpeuut,L
87-bpeuun,M
88-bpeuun,F
89-bpook,L
90-bpook,H
91-bpoong,F
92-bpoop,H
93-bpoom,L
94-bpoom,F
95-bpuu,M
96-bpuu,L
97-bpuut,L
98-bpuun,M
99-bpeh,F
100-bpeh,R
101-bpehk,H
102-bpaehng,M
103-bpehng,F
104-bpet,L
105-bpen,M
106-bprohk,L
107-bprohng,M
108-bprohn,M
109-bprohp,L
110-bpra,L
111-bpraeht,L
112-bpraehm,M
113-bpruuhr,M
114-bpruh,L
115-bpraw,L
116-bplaeh,M
117-bplehng,L
118-bplaw,L
119-bpeh,H
120-bpaw,L
121-bpeerk,L
122-bpeert,L
123-bpeern,L
124-bpeerp,L
125-bpeern,F
126-bpaae,M
127-bpaaeng,F
128-bpaaet,L
129-bpaaet,H
130-bpaaen,F
131-bpaaep,H
132-bpraae,M
133-bpraaeng,M
134-bpraeng,L
135-bplaae,M
136-bplaaek,L
137-bplaaeng,M
138-bplaaen,M
139-bpae,H
140-bpoxh,H
141-bpoxhng,L
142-bpoxhng,F
143-bpoxhp,H
144-bproxhng,L
145-bproxht,L
146-bpo,H
147-bpohk,L
148-bpohng,M

149-dtaaw,M
150-dtwat,L
151-dtaaw,L
152-dtaaw,F
153-dtaawk,L
154-dtaawk,H
155-dtaawng,M
156-dtaawng,F
157-dtaawng,H
158-dtaawt,L
159-dtaawn,M
160-dtaawn,F
161-dtaawp,L
162-dtaawm,M
163-dtaawm,L
164-dtaawm,F
165-dta,L
166-dtak,L
167-dtang,M
168-dtang,L
169-dtang,F
170-dtang[k],M
171-dtat,L
172-dtan,M
173-dtap,L
174-dtoht,L
175-dtohn,M
176-dtohn,F
177-dtaa,M
178-dtaak,L
179-dtaang,L
180-dtaan,M
181-dtaan,F
182-dtaap,L
183-dti,L
184-dting,M
185-dting,L
186-dtit,L
187-dtim,R
188-dtee,M
189-dtee,L
190-dtee,R
191-dteen,M
192-dteep,L
193-dteuk,L
194-dteung,M
195-dteuut,L
196-dteuun,L
197-dteuun,F
198-dteuu,F
199-dteuu,H
200-dteuu,R
201-dtoo,L
202-dtoo,H
203-dtook,H
204-dtoong,M
205-dtoon,M
206-dtoon,L
207-dtoon,R
208-dtoom,L
209-dtoom,F
210-dtuu,L
211-dtuu,F
212-dtuut,L
213-dtuup,L
214-dtuum,M
215-dteerp,L
216-dteerm,M
217-dtaae,L
218-dtaaek,L
219-dtaaeng,M
220-dtaeng,L
221-dtaaen,M
222-dtaaem,F
223-dtraae,M
224-dtae,L
225-dtoxh,M
226-dtoxh,F
227-dtoxhng,L
228-dtoxhng,F
229-dto,H
230-dtohp,L
231-dtohm,M
232-dtohm,F
233-dtrohng,M
234-dtrohm,M
235-dtraawk,L
236-dtraawng,M
237-dtrang,M
238-dtrap,L
239-dtrat,L
240-dtraa,M
241-dtraat,L
242-dtraap,L
243-dtree,M
244-dtreuk,L
245-dtreung,M
246-dtreum,M
247-dtroo,L
248-dtroot,L
249-dtruu,M
250-dtruu,L
251-dtohk,L
252-dtohng,M
253-dtohng,R
254-baat[r],L
255-baan,M
256-baan,F
257-baap,L
258-baa[r],M
259-baa[r]f,L
260-baat[k],L
261-bi,L
262-bik,H
263-bit,L
264-bin,M
265-bin,L
266-bin[l],M
267-bee,M
268-bee,F
269-beep,L
270-beuk,L
271-beung,M
272-beung,F
273-beum,F
274-beuu,F
275-book,L
276-book,H
277-boong,F
278-boon,M
279-boot[r],L
280-boon,R
281-boop,L
282-boom,M
283-boom,R
284-braehk,L
285-blaehs,L
286-baeht,L
287-buuhr,F
288-buuhr,M
289-buh,L
290-buh,H
291-beh,L
292-baw,L
293-beerk,L
294-beerng,L
295-beert,L
296-beerm,F
297-beern,M
298-baae,M
299-baaek,L
300-baeng,L
301-baaeng[k],M
302-baaen,M
303-baaep,L
304-bohm,L
305-breef,L
306-bruus,H
307-blawk,L
308-baaw,M
309-baaw,L
310-baaw,F
311-baaw,H
312-baawk,L
313-baawng,F
314-baawng,H
315-baawt,L
316-baawn,M
317-baawn,L
318-baawn[n],M
319-baawp,L
320-baawp,H
321-baawm[p],M
322-baaw[r]d,L
323-baawn[l],M
324-ba,H
325-bak,L
326-bang,M
327-bang,F
328-baa,L
329-baa,F
330-baak,L
331-baang,M
332-baang,L
333-baang,F
334-bohk,L
335-bohng,L
336-boht,L
337-bohn,M
338-bohn,L
339-daaw,M
340-doon,M
341-doon,F
342-doom,M
343-doom,L
344-doon[y],M
345-duu,M
346-duut,L
347-dek,L
348-dehng,F
349-det,L
350-daeht,L
351-deht,H
352-daehn,M
353-dehn,L
354-daehf,L
355-daehn[l],M
356-duuhr,F
357-duuhr,R
358-deh,L
359-daw,L
360-deern,M
361-deern,F
362-deerm,M
363-daae,L
364-daaek,L

365-daaeng,M
366-daaet,L
367-daaen,M
368-daen,L
369-dae,L
370-doxh,M
371-doxh,L
372-doxhng,L
373-doxht,L
374-doxhn,M
375-do,H
376-daawk,L
377-dawk,L
378-daawng,M
379-daawt,L
380-dawt,L
381-daawn,M
382-daawm,M
383-daawm,F
384-daawn[l],M
385-daa,M
386-daa,L
387-daa,R
388-daang,L
389-daat,L
390-daan,M
391-daan,L
392-daan,F
393-daap,L
394-di,L
395-dik,L
396-dik,H
397-ding,L
398-din,M
399-din,F
400-deen,M
401-deuk,L
402-deung,M
403-deung,L
404-deum,L
405-deuun,L
406-deuum,L
407-deuu,F
408-doo,L
409-dook,L
410-doong,F
411-doot,L
412-dohk,L
413-dohng,M
414-dohn,F
415-dohm,M
416-dohn,M
417-thaaw,R
418-thaan,R
419-thaaw,M
420-thoxh,R
421-thoxhng,R
422-thoxhp,L
423-thoxhm,R
424-thohm,L
425-thaaw,L
426-thaawk,L
427-thaawt,L
428-thaawn,R
429-thaawm,L
430-tha,L
431-thak,L
432-thang,R
433-that,L
434-than,R
435-than,L
436-thap,L
437-thaa,F
438-thaang,R
439-thaang,L
440-thaat,L
441-thaan,L
442-thin,L
443-thee,L
444-theep,L
445-theuk,L
446-theung,R
447-theuu,R
448-thoong,R
449-thoon,R
450-thuu,R
451-thuuk,L
452-thaehn,R
453-thuh,L
454-thaw,L
455-theerk,L
456-theert,L
457-theern,R
458-thaaep,L
459-thaaem,R
460-thohk,L
461-thoht,L
462-thohm,R
463-thap,H
464-that[n],H
465-thaa,M
466-thaa,H
467-thaak,F
468-thaang,M
469-thaan,M
470-than,F
471-thaap,F
472-thaa[r],M
473-thaat,F
474-thing,H
475-thit,H
476-thin,M
477-thip,H
478-thip[y],H
479-thim,M
480-thim,F
481-thee,M
482-thee,F
483-theem,M
484-theuk,H
485-theung,F
486-theung,H
487-theup,H
488-theum,M
489-theum,F
490-theuu,F
491-thoo,H
492-thook,H
493-thook[k],H
494-thoong,F
495-thoot,H
496-thoon,M
497-thoon,F
498-thoop,H
499-thoom,M
500-thoom,F
501-thoom,H
502-thuu,M
503-thuu,F
504-thuut,F
505-thuun,M
506-thuum,M
507-thaeh,M
508-theh,F
509-thaehk,F
510-thet,H
511-thaehp,F
512-thaehm[s],M
513-thraeht,F
514-theerng,M
515-theerng,F
516-theert,F
517-theern,M
518-thaae,H
519-thaaeng,M
520-thaaeng,F
521-thaaeng,H
522-thaeng[k],M
523-thaaen,M
524-thaaen,F
525-thaaep,F
526-thaep,H
527-thraaek,F
528-thraek,H
529-thae,H
530-thoxh,M
531-thoxht,F
532-thoht,H
533-thaaw,F
534-thaaw,H
535-thaawng,M
536-thaawng,F
537-thaawng,H
538-thaawt,F
539-thaawt,H
540-thaawn,M
541-thaawn,F
542-thaawp,H
543-thaawm,M
544-thak,H
545-thang,F
546-thang,H
547-than[t],M
548-that,H
549-than,M
550-than[d],M
551-thohn,M
552-thohn,H
553-thohn[d],M
554-thohp,H
555-than[m],M
556-than[w],M
557-thuup,F
558-thuuhr,M
559-thoxh,F
560-tha,H
561-thohng,M
562-phin,R
563-phee,R
564-pheung,L
565-pheung,F
566-pheuun,R
567-pheuun,L
568-phoo,L
569-phoot,L
570-phuu,F
571-phuuk,L
572-phaeh,R
573-phaehng,R
574-phet,L
575-phehn,L
576-phluuhr,R
577-pheern,R
578-phaae,L
579-phaaeng,R
580-phaaet,L

581-phaaen,R
582-phaen,L
583-phlaae,R
584-phlaaeng,R
585-phoxh,R
586-phoxhng,R
587-phoxhn,R
588-phloxh,L
589-phla,L
590-phlak,L
591-phlat,L
592-phlap,L
593-phlaan,R
594-phli,L
595-phloong,R
596-phloop,L
597-phaaw,R
598-phaawk,L
599-phaawng,R
600-phaawng,L
601-phaawn,L
602-phaawm,R
603-phak,L
604-phang,R
605-phat,L
606-phan,R
607-phap,L
608-phaa,R
609-phaa,L
610-phaa,F
611-phaak,L
612-phaang,R
613-phaat,L
614-phaan,L
615-phaan,R
616-phing,R
617-phit,L
618-phohng,R
619-phoht,L
620-phohm,R
621-phlohp,H
622-phlaawng,M
623-phlaawt,F
624-phlawm,M
625-phlak,F
626-phlang,F
627-phlang,H
628-phlat,H
629-phlan,M
630-phlap,H
631-phlaa,F
632-phlaang,M
633-phlaat,F
634-phlaan,F
635-phlik,H
636-phlee,M
637-phloo,H
638-phloong,F
639-phluu,M
640-phaaw,M
641-phaaw,F
642-phaaw,H
643-phaawk,F
644-phaawng,M
645-phaawng,H
646-phaawt,F
647-phaawn,M
648-phaawp,H
649-phak,H
650-phak[dr],H
651-phang,M
652-phat,H
653-phan,M
654-phan[ch],M
655-phan[t],M
656-phap,H
657-phat[dr],H
658-phaa,M
659-phaak,F
660-phaak[y],F
661-phaang,F
662-phaat,F
663-phaan,M
664-phing,M
665-phit[n],H
666-phin,M
667-phohp,H
668-phim[p],M
669-phit,H
670-phee,M
671-phee,F
672-pheet,F
673-phees,F
674-pheung,M
675-pheum,M
676-pheuut,F
677-pheuun,H
678-pheuun,M
679-phoo,H
680-phook,H
681-phoong,M
682-phoong,F
683-phoot,H
684-phoom,F
685-phuu,M
686-phuut,F
687-phuun,M
688-phehng,F
689-phaeht,F
690-phet,H
691-pheht,H
692-phen,M
693-phraw,H
694-phaehn,M
695-phleh,F
696-phlaehng,M
697-phlaeh[y],M
698-phlaw,H
699-phleerng,M
700-phleert,F
701-phleern,M
702-phuuhr,F
703-phuuhr,H
704-pheh,H
705-phaw,H
706-pheerk,F
707-pheerng,M
708-pheerng,F
709-pheern,F
710-pheerm,F
711-phaae,M
712-phaae,H
713-phaaeng,M
714-phaaeng,F
715-phaaet,F
716-phaaet[y],F
717-phaaen,M
718-phaaen,F
719-phraae,M
720-phaae[r],M
721-phraae,F
722-phraaeng,F
723-phlaaeng,M
724-phlaaem,M
725-phlaem,M
726-phae,H
727-phoxhk,F
728-phoxhng,M
729-phoxht,F
730-phoxhn,M
731-phoxhn,H
732-phroxhk,F
733-phroxhng,M
734-phloxhng,M
735-phloxhng,F
736-phloxhn,M
737-phoxht[d],F
738-phroht,H
739-phrohm,M
740-phan[k],M
741-phraawng,F
742-phraawng,H
743-phraawm,H
744-phra,H
745-phran,F
746-phraa,F
747-phraa,H
748-phraak,F
749-phraang,M
750-phraang,F
751-phraan,M
752-phrik,H
753-phohk,H
754-phohng,M
755-phohng[s],M
756-phoht[n],H
757-phohn,F
758-phohn,H
759-phat[r],H
760-phaap,F
761-phruu,M
762-phak[s],H

C.1.2. Example token list

256-baan,F,f5
537-thaawng,H,f5
70-bpaak,L,f5
177-dtaa,M,f5
432-thang,R,f5

C.1.3. Continuous token list

801-bping,F
802-baan,F
803-thoong,F
804-phaaw,F
805-phaap,F
806-dteuu,H
807-dto,H
808-thaawng,H
809-phae,H
810-phrik,H
811-bprap,L
812-dtap,L
813-beep,L
814-baaep,L
815-daap,L
816-bpuu,M
817-bpaa,M
818-bpeuun,M
819-dtaa,M
820-din,M
821-thaawn,R
822-thang,R
823-thoong,R
824-phee,R
825-phohng,R

C.1.4. Simplified token database

471-thaap,F
152-dtaaw,F
473-thaat,F
735-phloxhng,F
706-pheerk,F
130-bpaaen,F
538-thaawt,F
164-dtaawm,F
700-phleert,F
228-dtoxhng,F
721-phraae,F
160-dtaawn,F
437-thaa,F
760-phaap,F
676-pheuut,F
531-thoxht,F
708-pheerng,F
689-phaeht,F
410-doong,F
662-phaat,F
715-phaaet,F
714-phaaeng,F
732-phroxhk,F
722-phraaeng,F
470-than,F
702-phuuhr,F
748-phraak,F
718-phaaen,F
626-phlang,F
623-phlaawt,F
467-thaak,F
686-phuut,F
407-deuu,F
524-thaaen,F
464-that[n],H
756-phoht[n],H
683-phoot,H
691-pheht,H
690-phet,H
705-phaw,H
747-phraa,H
669-phit,H
738-phroht,H
665-phit[n],H
487-theup,H
526-thaep,H
477-thip,H
762-phak[s],H
262-bik,H
539-thaawt,H
537-thaawng,H
649-phak,H
478-thip[y],H
474-thing,H
628-phlat,H
809-phae,H
324-ba,H
475-thit,H
752-phrik,H
642-phaaw,H
726-phae,H
529-thae,H
306-bruus,H
657-phat[dr],H
552-thohn,H
534-thaaw,H
492-thook,H
635-phlik,H
431-thak,L
406-deuum,L
235-dtraawk,L
166-dtak,L
426-thaawk,L
427-thaawt,L
286-baeht,L
567-pheuun,L
368-daen,L
436-thap,L
364-daaek,L
430-tha,L
284-braehk,L
295-beert,L
293-beerk,L
212-dtuut,L
257-baap,L
193-dteuk,L
443-thee,L
109-bprohp,L
201-dtoo,L
239-dtrat,L
250-dtruu,L
118-bplaw,L
442-thin,L
460-thohk,L
244-dtreuk,L
588-phloxh,L
183-dti,L
440-thaat,L
455-theerk,L
603-phak,L
122-bpeert,L
238-dtrap,L
507-thaeh,M
113-bpruuhr,M
692-phen,M
362-deerm,M
540-thaawn,M
225-dtoxh,M
699-phleerng,M
105-bpen,M
670-phee,M
734-phloxhng,M
249-dtruu,M
95-bpuu,M
694-phaehn,M
530-thoxh,M
558-thuuhr,M
819-dtaa,M
651-phang,M
170-dtang[k],M
365-daaeng,M
246-dtreum,M
326-bang,M
517-theern,M
488-theum,M
233-dtrohng,M
133-bpraaeng,M
138-bplaaen,M
172-dtan,M
81-bpeen,M
556-than[w],M
162-dtaawm,M
733-phroxhng,M
696-phlaehng,M
191-dteen,M
112-bpraehm,M
572-phaeh,R
584-phlaaeng,R
577-pheern,R
457-theern,R
586-phoxhng,R
823-thoong,R
579-phaaeng,R
615-phaan,R
597-phaaw,R
585-phoxh,R
452-thaehn,R
387-daa,R
562-phin,R
573-phaehng,R
824-phee,R
587-phoxhn,R
822-thang,R
606-phan,R
462-thohm,R
616-phing,R
421-thoxhng,R
566-pheuun,R
608-phaa,R
563-phee,R

581-phaaen,R	620-phohm,R	612-phaang,R	450-thuu,R
604-phang,R	438-thaang,R	253-dtohng,R	
576-phluuhr,R	428-thaawn,R	432-thang,R	

APPENDIX D: COMPUTATIONAL MODEL CODE

D.1. RunModel.pl

```
#!/usr/bin/perl
my $datadir = 'E:\Family\Documents\jami\database\Thai';
my @tonedirs = qw(high mid low rising falling);
my %tones = (H, "high", M, "mid", L, "low", R, "rising", F, "falling");
print "Experiment number: ";
chomp(my $experiment = <>);
my $experimentname = 'Test'."$experiment"; ##Testing stage!!
mkdir "$experimentname";
$finalDatadir = 'Data'."$experiment"; ##Data directory separate from Experiment directories.
mkdir "$finalDatadir";
print "\nHow many trials to run?: ";
chomp(my $trials = <>);
print "\n \# of Samples per citation tone category (20 samples): ";
chomp(my $spert = <>); ##Usually set at 20 words per tone for a total of 100 words.
print "\n \# of Samples per continuous tone category (5 samples): ";
chomp(my $spertcont = <>); ##Usually set at 5 words per tone for a total of 25 words.
print "\n What frequency? (i.e., 2Hz 4Hz, \<5Hz, 10Hz, 13Hz, 20Hz\>, 65Hz): ";
chomp(my $res = <>); ##The input should be a number. Don't include the 'Hz' in your input.
my $decay = 0;
open(R,"<bin\\pretestexamples.txt") || warn "Tbere ain't no such file, see: $!\n";
chomp(my @exampleset = <R>);
close(R);
my $min = 80; ##The frequency range the model will cover is 240Hz.
my $max = 320;
my $range = $max - $min;
my $incr = $range/$res;
my @modelements = ($res, $decay, $min, $max, $range, $incr);
for $trial(1..$trials){
        print "\n\t\tTrial $trial\n";

##############################Makedirectories##############################################

        print "\nMaking Directories for experiment: $experimentname, trial: $trial\n";
        $trialdir = "$experimentname\\$trial";
        mkdir "$trialdir";
        $preDatadir = "$finalDatadir\\pretest";
        mkdir "$preDatadir";
                $preDataTrials = "$preDatadir\\$trial";
                mkdir "$preDataTrials";
        $postDatadir = "$finalDatadir\\posttest";
        mkdir "$postDatadir";
                $postDataTrials = "$postDatadir\\$trial";
                mkdir "$postDataTrials";
        $lists = "$trialdir\\lists";
        mkdir "$lists";
        $MatrixFiles = "$trialdir\\batch";
        mkdir "$MatrixFiles";
                $preTestMatrixFiles = "$MatrixFiles\\pretest";
                mkdir "$preTestMatrixFiles";
                $postTestMatrixFiles = "$MatrixFiles\\posttest";
                mkdir "$postTestMatrixFiles";
                $preModelMatrixFiles = "$MatrixFiles\\premodel";
                mkdir "$preModelMatrixFiles";
                $postModelMatrixFiles = "$MatrixFiles\\postmodel";
                mkdir "$postModelMatrixFiles";
        $ObservationFiles = "$trialdir\\observations";
        mkdir "$ObservationFiles";
                $preObservationFiles = "$ObservationFiles\\pretest";
                        &toneDirs($preObservationFiles);
                $postObservationFiles = "$ObservationFiles\\posttest";
                        &toneDirs($postObservationFiles);
        $ModelFiles = "$trialdir\\models";
        mkdir "$ModelFiles";
```

```
		$RandModels = "$ModelFiles\\randmodels";
		mkdir "$RandModels";
		$preTestModels = "$ModelFiles\\pretest";
		mkdir "$preTestModels";
		$postTestModels = "$ModelFiles\\posttest";
		mkdir "$postTestModels";
			$postTestSubModels = "$postTestModels\\submodels";
				&toneDirs($postTestSubModels);


####################################Observations#################################################

	print "\nGenerating pre-train test observations, experiment: $experimentname, trial:
$trial\n";
	chomp(@preTestList = `perl bin/randomSampler.pl $spert $spertcont`);
	open LOGS, ">$lists\\preList.list" || warn "Can't print to $lists\\preList.list, see:
$!\n";
	print LOGS @preTestList;
	close(LOGS);
	for $i (0..$#preTestList){
		@line = split(/,/,$preTestList[$i]);
		$file = "$datadir\\$line[2]\\$line[0].wav";
		$outMatrixFile = "$preTestMatrixFiles\\$line[0]-$line[2]-$i";
		$MatrixFile = &Autocorrelation($file,$outMatrixFile);
		$outObserveFile = "$preObservationFiles\\$tones{$line[1]}\\$line[0]-$line[2]-
$i.csv";
		&MakeObservations($MatrixFile,$outObserveFile,$line[2]);
	};
	print "\nGenerating training list, experiment: $experimentname, trial: $trial\n";
	$trspert = $spert + 3;
	$trspertcont = $spertcont + 2 unless $spertcont == 0;
	chomp(@trainList = `perl bin/randomSampler.pl $trspert $trspertcont`);
	open LOGS, ">$lists\\trainList.list" || warn "Can't print to $lists\\trainList.list, see:
$!\n";
	print LOGS @trainList;
	close(LOGS);
	print "\nGenerating post-training observations, experiment: $experimentname, trial:
$trial\n";
	chomp(@postTestList = `perl bin/randomSampler.pl $spert $spertcont`);
	open LOGS, ">$lists\\postList.list" || warn "Can't print to $lists\\postList.list, see:
$!\n";
	print LOGS @postTestList;
	close(LOGS);
	for $i (0..$#postTestList){
		@line = split(/,/,$postTestList[$i]);
		$file = "$datadir\\$line[2]\\$line[0].wav";
		$outMatrixFile = "$postTestMatrixFiles\\$line[0]-$line[2]-$i";
		$MatrixFile = &Autocorrelation($file,$outMatrixFile);
		$outObserveFile = "$postObservationFiles\\$tones{$line[1]}\\$line[0]-$line[2]-
$i.csv";
		&MakeObservations($MatrixFile,$outObserveFile,$line[2]);
	};

#################################Models#############################################################

	print "\nGenerating models for experiment: $experimentname, trial: $trial\n";
	foreach(@tonedirs){
		print "\n\tgenerating random models for experiment: $experimentname, trial:
$trial, tone: $_\n";
		@A1 = `perl bin/MakeRandTran.pl $incr`;
		open RANDTRANS, ">$RandModels\\randtrans$_.csv" || warn "randtrans$_.csv, $!\n";
		print RANDTRANS @A1;
		close(RANDTRANS);
		@B1 = `perl bin/MakeRandEmis.pl $incr`;
		open RANDEMIS, ">$RandModels\\randemis$_.csv" || warn "randemis$_.csv, $!\n";
		print RANDEMIS @B1;
		close(RANDEMIS);
		@pi1 = `perl bin/MakeRandPis.pl $incr`;
		open RANDPIS, ">$RandModels\\randpis$_.csv" || warn "randpis$_.csv, $!\n";
```

```
                print RANDPIS @pi1;
                close(RANDPIS);
                print "\n\tgenerating pre-training models for experiment: $experimentname, trial:
$trial, tone: $_\n";
                for $i (0..$#exampleset){
                        @line = split(/,/,$exampleset[$i]);
                        if ($tones{$line[1]} eq $_) {
                                @majorseq = ();
                                @pi = ();
                                @A = ();
                                @B = ();
                                $preFile = "bin\\examples\\$line[2]\\$line[0].wav";
                                $preOutFile = "$preModelMatrixFiles\\$line[0]-$line[2]-$i";
                                $preModelMatrixFile = &Autocorrelation($preFile,$preOutFile);
                                @preFreqs = ();
                                @preFreqs = &makedatasets($preModelMatrixFile);
                                @seq = ();
                                @seq = ($preFreqs[0], $preFreqs[$#preFreqs * 0.1],
$preFreqs[$#preFreqs * 0.3], $preFreqs[$#preFreqs * 0.5], $preFreqs[$#preFreqs * 0.7],
$preFreqs[$#preFreqs * 0.9]);
                                };
                                for $exset (2..5){
                                        push(@majorseq,"$seq[$exset - 1],$seq[$exset],f5\n");
                                };
                                @A = `perl bin/MakeTrans.pl @modelements @A1 @majorseq`;
                                @A1 = ();
                                @A1 = @A;
                                @B = `perl bin/MakeEmis.pl @modelements @B1 @majorseq`;
                                @B1 = ();
                                @B1 = @B;
                                @pi = `perl bin/MakePis.pl @modelements $seq[0] @pi1`;
                                @pi1 = ();
                                @pi1 = @pi;
                        };
                };
                open PRETRANS, ">$preTestModels\\trans$_.csv" || warn
"$preTestModels\\trans$_.csv, $!\n";
                print PRETRANS @A1;
                close(PRETRANS);
                open PREEMIS, ">$preTestModels\\emis$_.csv" || warn "$preTestModels\\emis$_.csv,
$!\n";
                print PREEMIS @B1;
                close(PREEMIS);
                open PREPI, ">$preTestModels\\init$_.csv" || warn "$preTestModels\\init$_.csv,
$!\n";
                print PREPI @pi1;
                close(PREPI);

                print "\n\tgenerating training models for experiment: $experimentname, trial:
$trial, tone: $_\n";

                my @majorseq = ();
                my @A = ();
                my @B = ();

                for $i (0..$#trainList){
                        @line = split(/,/,$trainList[$i]);
                        if ($tones{$line[1]} eq $_) {
                                my @pi = ();
                                $trainFile = "$datadir\\$line[2]\\$line[0].wav";
                                $trainOutFile = "$postModelMatrixFiles\\$line[0]-$line[2]-$i";
                                $postModelMatrixFile = &Autocorrelation($trainFile,$trainOutFile);
                                @trainFreqs = ();
                                @trainFreqs = &makedatasets($postModelMatrixFile);
                                @seq = ();
                                @seq = ($trainFreqs[0], $trainFreqs[$#trainFreqs * 0.1],
$trainFreqs[$#trainFreqs * 0.3], $trainFreqs[$#trainFreqs * 0.5], $trainFreqs[$#trainFreqs *
0.7], $trainFreqs[$#trainFreqs * 0.9]);
                                for $set (2..5){
```

```
                                        push(@majorseq,"$seq[$set - 1],$seq[$set],$line[2]\n");
                                };
                                @pi = `perl bin/MakePis.pl @modelements $seq[0] @pi1`;
                                undef @pi1;
                                @pi1 = @pi;

                        };
                };

                @A = `perl bin/MakeTrans.pl @modelements @A1 @majorseq`;
                undef @A1;
                @A1 = @A;
                @B = `perl bin/MakeEmis.pl @modelements @B1 @majorseq`;
                undef @B1;
                @B1 = @B;

                open POSTTRANS, ">$postTestModels\\trans$_.csv" || warn
"$postTestModels\\trans$_.csv, $!\n";
                print POSTTRANS @A1;
                close(POSTTRANS);
                open POSTEMIS, ">$postTestModels\\emis$_.csv" || warn
"$postTestModels\\emis$_.csv, $!\n";
                print POSTEMIS @B1;
                close(POSTEMIS);
                open POSTPI, ">$postTestModels\\init$_.csv" || warn "$postTestModels\\init$_.csv,
$!\n";
                print POSTPI @pi1;
                close(POSTPI);
        };

##################################################Testing#######################################

        print "\nTesting models for experiment: $experimentname, trial: $trial\n";
        foreach(@tonedirs){
                $matexpertest = "experiment=\'$experimentname\'";
                $matexperdata = "experdata=\'$finalDatadir\'";
                $mattest = "test=\'pretest\'";
                $himat = "tone=\'$_\'";
                $mattrial = "trial=\'$trial\'";
                print "\n\tHMM pre-test for experiment: $experimentname, trial: $trial, tone:
$_\n";
                system `matlab -nodesktop -nosplash -r "$matexpertest, $matexperdata, $mattrial,
$mattest, $himat, MaxLogpsec"`;
                print "\n\tHMM post-test for experiment: $experimentname, trial: $trial, tone:
$_\n";
                $mattest = "test=\'posttest\'";
                system `matlab -nodesktop -nosplash -r "$matexpertest, $matexperdata, $mattrial,
$mattest, $himat, MaxLogpsec"`;
        };
};

##################################################ConfusionMatrix#################################

print "\n\t\t\#\#\#DONE Experiment: $experimentname\#\#\#\n";

##################################################Subroutines#####################################

sub clearFolder {
        mkdir "$_[0]";
        opendir(R,$_[0]) || warn "ore else, $!";
        @oldfiles = readdir(R);
        closedir(R);
        foreach(@oldfiles){
                unlink("$_[0]\\$_");
        };
};

sub toneDirs{
        @tonedirs = (high, mid, low, rising, falling);
```

```
	mkdir "$_[0]";
	foreach(@tonedirs){
		$dirtydir = "$_[0]/$_";
		mkdir "$dirtydir";
		&clearFolder($dirtydir);
	};
};

sub Autocorrelation{
	$input = $_[0];
	$outname = $_[1];
	system `start praatcon.exe pitch2matrix-02.praat $input $outname`;
	$outname = $outname.".Matrix";
	return $outname;
};

sub makedatasets{
	open(R,"<$_[0]") || warn "Can't open $_[0], $!\n";
	chomp(@a = <R>);
	close(R);
	@allfreqs = ();
	for $x (15..$#a) {
		@freq = split(/= /,$a[$x]);
		$freq = substr($freq[1],0,-2);
		$freq = 0 if $freq eq "";
		push(@allfreqs,$freq) unless $freq eq 0;
	};
	return @allfreqs;
};

sub MakeObservations{
	my %means = (f5, 220.4888, f6, 166.9435, f7, 205.0879, f8, 178.3145, f9, 235.8949, f10,
229.6710);
	my %stds = (f5, 30.2785, f6, 20.2363, f7, 28.1023, f8, 22.6342, f9, 30.5193, f10,
32.6692);
	@ofreq = ();
	@ofreq = &makedatasets($_[0]);
	open(S,">$_[1]") || warn "Oh suck! I can't write to anything, see: $!\n";
	@seq = ();
	@seq = ($ofreq[$#ofreq * 0.1], $ofreq[$#ofreq * 0.3], $ofreq[$#ofreq * 0.5],
$ofreq[$#ofreq * 0.7], $ofreq[$#ofreq * 0.9]);
	@normalseq = ();
	for $k(0..$#seq){
		$zscore = ($seq[$k] - $means{$_[2]})/$stds{$_[2]};
		if ($zscore < -1.5){
			$tonenum = "1";
		}
		elsif ($zscore >= -1.5 && $zscore < -0.5){
			$tonenum = "2";
		}
		elsif ($zscore >= -0.5 && $zscore < 0.5){
			$tonenum = "3";
		}
		elsif ($zscore >= 0.5 && $zscore < 1.5){
			$tonenum = "4";
		}
		else{
			$tonenum = "5";
		};
		push(@normalseq,"$tonenum,");
	};
	print S @normalseq;
	close(S);
};
```

D.2. randomSampler.pl

```
#!/usr/bin/perl

use List::Util qw(shuffle sum);

my $spert = $ARGV[0];
my $spertcont = $ARGV[1];
my @speakerList = (f6,f7,f8,f9,f10); ##If there is a different set of speakers.

open Q, "<bin\\toneList.txt" || warn "Could not open index-file, see: $!\n";
chomp(my @toneList = <Q>);
close(Q);
open R, "<bin\\toneListCont.txt" || warn "Could not open index-file, see: $!\n";
chomp(my @toneListCont = <R>);
close(R);

@citList = ();
@citList = &randomShuffle($spert,758,\@toneList); ##for citation words only. Sample count, number
in list, and list.
@contList = ();
@contList = &randomShuffle($spertcont,24,\@toneListCont);  ##for continuous words only.
@masterList = ();
@masterList = shuffle(@citList,@contList);
@finalList = ();
@finalList = &makeSpeakerToneList(\@masterList,\@speakerList);
print @finalList;

##########################################################################################
##########################################################################################

sub randomShuffle {
	@pend = @{$_[2]};
	@word = ();
	@shuffled = ();
	%toneVal = (L, 0, H, 0, M, 0, R, 0, F, 0);
	$totaltVal = 0;
	while($totaltVal != ($_[0] * 5)){
		$roll3 = int(rand $_[1]);
		@line = split(/\,/,$pend[$roll3]);
		unless($toneVal{$line[1]} >= $_[0]){
			$toneVal{$line[1]}++;
			push(@word,$pend[$roll3]);
		};
		$totaltVal = sum values %toneVal;
		@shuffled = shuffle(@word);
	};
	return @shuffled;
};

sub makeSpeakerToneList{
	@made = ();
	@sample = @{$_[0]};
	@speakers = @{$_[1]};
	foreach(@sample){
		$speaker = $speakers[int(rand 5)];
		$_ =~ tr/\[\]/\_\_/ unless $speaker eq "f6";
		push(@made,"$_,$speaker\n");
	};
	return @made;
};
```

D.3. MakeRandTran.pl

```
#!/usr/local/bin/perl

my $incr = $ARGV[0];
my @aa = ();
my $incrl = $incr - 1;

for $i (1..$incr){
        @totali = ();
        @ai = ();
        $total = 0;
        for $j (1..$incr){
                $num = int(rand(10));
                $total = $total + $num;
                push(@totali,$num);
        };
        for $j (1..$incrl){
                $value = $totali[$j - 1] / $total;
                push(@ai,"$value,");
        };
        $value = $totali[$#totali] / $total;
        push(@ai,"$value\n");
        push(@aa,@ai);
};

print @aa;
```

D.4. MakeRandEmis.pl

```
#!/usr/local/bin/perl

my $incr = $ARGV[0];
my $incrl = $incr - 1;
my @bb = ();

for $i (1..$incr){
        @totali = ();
        @bi = ();
        $total = 0;
        for $j (1..5){
                $num = int(rand(10));
                $total = $total + $num;
                push(@totali,$num);
        };
        for $j (1..4){
                $value = $totali[$j - 1] / $total;
                push(@bi,"$value,");
        };
        $value = $totali[$#totali] / $total;
        push(@bi,"$value\n");
        push(@bb,@bi);
};

print @bb;
```

D.5. MakeRandPis.pl

```
#!/usr/local/bin/perl

my $incr = $ARGV[0];
my $incrl = $incr - 1;

for $i (1){
	@totalpi = ();
	@pi = ();
	$total = 0;
	for $j (1..$incr){
		$num = int(rand(10));
		$total = $total + $num;
		push(@totalpi,$num);
	};
	for $j (1..$incrl){
		$value = $totalpi[$j - 1] / $total;
		push(@pi,"$value,");
	};
	$value = $totalpi[$#totalpi] / $total;
	push(@pi,"$value\n");
};
print @pi;
```

D.6. MakeTrans.pl

```
#!/usr/bin/perl

my $res = $ARGV[0];
my $decay = $ARGV[1];
my $min = $ARGV[2];
my $max = $ARGV[3];
my $range = $ARGV[4];
my $incr = $ARGV[5];
my @A1 = @ARGV[6..17];
my @seq = @ARGV[18..$#ARGV];

my @A =();

for $ii (1..$incr){
	@trans = ();
	$ibot = $min + ($res * ($ii - 1));
	$itop = $min + ($res * $ii);
	@transi = ();
	($total, @trans) = &stateobservationstrans($incr, \@seq, $ibot, $itop, $min, $res,
$A1[$ii-1]);
	$total++ if $total == 0;
	@transi = &statetransitionmatrix($total, \@trans);
	push(@A,@transi);
};

print @A;

##########################################################################################

sub stateobservationstrans{
	$incrj = $_[0];
	@append = @{$_[1]};
	@cj = split(/\,/,$_[6]);
	$total = 0;
	@transx = ();
	for $jj (1..$incrj){
		$observations = 0;
		$jbot = $_[4] + ($_[5] * ($jj - 1));
```

```
                $jtop = $_[4] + ($_[5] * $jj);
                foreach (@append){
                        @line = split(/\,/,$_);
                        $s1 = 0;
                        $s2 = 0;
                        $s1 = 1 if ($line[0] >= $_[2] && $line[0] < $_[3]);
                        $s2 = 1 if ($line[1] >= $jbot && $line[1] < $jtop);
                        $observable = $s1 + $s2;
                        $observations++ if $observable == 2;
                };
                $tran = $observations + $cj[$jj-1];
                push(@transx,$tran);
        };
        my $total = ();
        $total += $_ foreach @transx;
        return ($total, @transx);
};

sub statetransitionmatrix{
        @transi = ();
        @model = @{$_[1]};
        $xxl = $#model - 1;
        for $xx (0..$xxl){
                $tri = $model[$xx]/$_[0];
                push(@transi,"$tri,");
        }
        $tri = $model[$#model]/$_[0];
        push(@transi,"$tri\n");
        return @transi;
};
```

D.7. MakeEmis.pl

```
#!/usr/bin/perl

my $res = $ARGV[0];
my $decay = $ARGV[1];
my $min = $ARGV[2];
my $max = $ARGV[3];
my $range = $ARGV[4];
my $incr = $ARGV[5];
my @B1 = @ARGV[6..17];
my @seq = @ARGV[18..$#ARGV];


my @B =();

for $i(1..$incr){
        @emis = ();
        $ibot = $min + ($res * ($i - 1));
        $itop = $min + ($res * $i);
        $emits = 0;
        ($emits, @emis) = &stateobservationsemis($incr, \@seq, $ibot, $itop, $min, $res, $B1[$i-
1]);
        $emits++ if $emits == 0;
        @emisi = &stateemisionmatrix($emits, \@emis);
        push(@B,@emisi);
};

print @B;

##########################################################################################

sub stateobservationsemis{
        my %means = (f5, 220.4888, f6, 166.9435, f7, 205.0879, f8, 178.3145, f9, 235.8949, f10,
229.6710);
```

```
        my %stds = (f5, 30.2785, f6, 20.2363, f7, 28.1023, f8, 22.6342, f9, 30.5193, f10,
32.6692);
        @cemit = split(/\,/,$_[6]);
        @append = @{$_[1]};
        @emis = ();
        $LL = 0; ## Tone level 1
        $L = 0; ## level 2
        $M = 0; ## level 3
        $H = 0; ## level 4
        $HH = 0; ## level 5

        foreach (@append){
                @line = split(/\,/,$_);
                if($line[1] >= $_[2] && $line[1] < $_[3]){
                        $zscore = ($line[1] - $means{$line[2]})/$stds{$line[2]};
                        $LL++ if $zscore < -1.5;
                        $L++ if $zscore >= -1.5 && $zscore < -0.5;
                        $M++ if $zscore >= -0.5 && $zscore < 0.5;
                        $H++ if $zscore >= 0.5 && $zscore < 1.5;
                        $HH++ if $zscore >= 1.5;
                };
        };

        $LL = $LL + $cemit[0];
        $L = $L + $cemit[1];
        $M = $M + $cemit[2];
        $H = $H + $cemit[3];
        $HH = $HH + $cemit[4];
        $emit = "$LL,$L,$M,$H,$HH";
        $emits = $LL + $L + $M + $H + $HH;
        push(@emis,$emit);
        return($emits,@emis);
};

sub stateemisionmatrix{
        @emisi = ();
        foreach(@{$_[1]}){
                @emi = split(/\,/,$_);
                $emLLi = $emi[0]/$_[0];
                $emLi = $emi[1]/$_[0];
                $emMi = $emi[2]/$_[0];
                $emHi = $emi[3]/$_[0];
                $emHHi = $emi[4]/$_[0];
                $emisions = "$emLLi,$emLi,$emMi,$emHi,$emHHi\n";
                push(@emisi,$emisions);
        };
        return @emisi;
};
```

D.8. MakePis.pl

```
#!/usr/bin/perl

my $res = $ARGV[0];
my $min = $ARGV[2];
my $incr = $ARGV[5];
my $inifreq = $ARGV[6];
my @pill = @ARGV[7..$#ARGV];
my @pi1 = split(/,/,$pill[0]);
my @pi =();
push(@pi,$inifreq);
my $sN = $incr + 1;
my @pis = ();

for $ipi(2..$sN){
        $ident = 0;
        foreach(@pi){
                $xn = ($_ - $min)/$res;
                if($xn < $ipi && $xn >= ($ipi - 1)){
                        $ident++;
                };
        };
        $ident = $ident + $pi1[$ipi - 2];
        push(@pis,$ident);
};

$tpi = 0;
$tpi += $_ foreach @pis;
$tpi++ if $tpi == 0;

$xpisl = $#pis - 1;

for $xpi (0..$xpisl){
        $initpi = $pis[$xpi]/$tpi;
        push(@S0,"$initpi,");
};
$initpi = $pis[$#pis]/$tpi;
push(@S0,"$initpi");

print @S0;
```

D.9. pitch2matrix.praat

```
form foo
        word filename ""
        word outname ""
endform

Read from file... 'filename$'
To Pitch (ac)... 0 75 15 yes 0.03 0.6 0.15 0.35 0.14 350
Smooth... 10
To Matrix
Save as text file... 'outname$'.Matrix

Quit
```

D.10. MaxLogpsec.m

```
modfiles = fullfile(experiment,trial,'models',test);

transhigh = csvread(strcat(modfiles,'\transhigh.csv'));
transmid = csvread(strcat(modfiles,'\transmid.csv'));
translow = csvread(strcat(modfiles,'\translow.csv'));
transrising = csvread(strcat(modfiles,'\transrising.csv'));
transfalling = csvread(strcat(modfiles,'\transfalling.csv'));

emishigh = csvread(strcat(modfiles,'\emishigh.csv'));
emismid = csvread(strcat(modfiles,'\emismid.csv'));
emislow = csvread(strcat(modfiles,'\emislow.csv'));
emisfalling = csvread(strcat(modfiles,'\emisfalling.csv'));
emisrising = csvread(strcat(modfiles,'\emisrising.csv'));

inithigh = csvread(strcat(modfiles,'\inithigh.csv'));
transhigh_hat = [ 0 inithigh ; zeros(size(transhigh,1),1) transhigh ];
emishigh_hat = [zeros(1,size(emishigh,2)); emishigh ];
initmid = csvread(strcat(modfiles,'\initmid.csv'));
transmid_hat = [ 0 initmid ; zeros(size(transmid,1),1) transmid ];
emismid_hat = [zeros(1,size(emismid,2)); emismid ];
initlow = csvread(strcat(modfiles,'\initlow.csv'));
translow_hat = [ 0 initlow ; zeros(size(translow,1),1) translow ];
emislow_hat = [zeros(1,size(emislow,2)); emislow ];
initfalling = csvread(strcat(modfiles,'\initfalling.csv'));
transfalling_hat = [ 0 initfalling ; zeros(size(transfalling,1),1) transfalling ];
emisfalling_hat = [zeros(1,size(emisfalling,2)); emisfalling ];
initrising = csvread(strcat(modfiles,'\initrising.csv'));
transrising_hat = [ 0 initrising ; zeros(size(transrising,1),1) transrising ];
emisrising_hat = [zeros(1,size(emisrising,2)); emisrising ];

myObsrv = fullfile(experiment,trial,'observations',test,tone);
filePattern = fullfile(myObsrv,'*.csv');
obsrvFiles = dir(filePattern);

for k = 1:size(obsrvFiles,1)
    baseFileName = obsrvFiles(k).name;
    fullFileName = fullfile(myObsrv,baseFileName);
    fprintf('Now reading %s\n',fullFileName);
    seq = importdata(fullFileName);
    tic;
    [pstates01,logpsec01] = hmmdecode(seq,transhigh_hat,emishigh_hat);
    logpsec(k,1) = logpsec01;
    [pstates02,logpsec02] = hmmdecode(seq,transmid_hat,emismid_hat);
    logpsec(k,2) = logpsec02;
    [pstates03,logpsec03] = hmmdecode(seq,translow_hat,emislow_hat);
    logpsec(k,3) = logpsec03;
    [pstates04,logpsec04] = hmmdecode(seq,transfalling_hat,emisfalling_hat);
    logpsec(k,4) = logpsec04;
    [pstates05,logpsec05] = hmmdecode(seq,transrising_hat,emisrising_hat);
    logpsec(k,5) = logpsec05;
    seqIDend(k,1) = toc;
end

Datadir = fullfile(experdata,test,trial);
saveFile = strcat(Datadir,'\data',tone,'.csv');
saveFiletime = strcat(Datadir,'\time',tone,'.csv');

A = dataset(logpsec);
export(A,'file',saveFile,'Delimiter',',');

B = dataset(seqIDend);
export(B,'file',saveFiletime,'Delimiter',',');

clear;

exit;
```

D.11. cmaxModel.pl

```
#!/usr/bin/perl

use List::Util 'max';

####################
##global variables##
####################

my %tonedir = (0, H, 1, M, 2, L, 3, F, 4, R);
my %tonelabels = ('datahigh.csv', H, 'datamid.csv', M, 'datalow.csv', L, 'datarising.csv', R,
'datafalling.csv', F);


##################
##Standard Input##
##################

print "\n(pretest) or (posttest)?: ";
chomp($whattheysay = <>);
if($whattheysay eq 'pretest'){
	$datafolder = "pretest"
}
elsif($whattheysay eq 'posttest'){
	$datafolder = "posttest"
}
else{
	warn "oops!";
	die;
};

opendir(R,$datafolder) || warn "ore else $!";
@filedir02 = readdir(R);
closedir(R);
@filedir02 = @filedir02[2..$#filedir02];

open(S,">$datafolder.csv") || warn "No such luck, $!";

foreach(@filedir02){
	$rawdatafiles = "$datafolder/$_";
	opendir(R,$rawdatafiles) || warn "Oh Dagnabbit, there is $!\n";
	@rawdatafile = readdir(R);
	closedir(R);
	@rawdatafile = grep(/data/,@rawdatafile);
	print S "\n$_\n";
	print S ",H,M,L,R,F\n";
	foreach(@rawdatafile){
		$getfiles = "$rawdatafiles/$_";
		open(R,"<$getfiles") || warn "Oh Dagnabbit, there is $!\n";
		chomp(@a = <R>);
		close(R);
		%count = (H, 0, M, 0, L, 0, R, 0, F, 0);

		for $i(1..$#a){
			@data = split(/\,/,$a[$i]);
			$max = max(@data);
			for $j(0..$#data){
				if($data[$j] eq $max){
					$key = $tonedir{$j};
				};
			};
			$count{$key}++;
		};

		print S "$tonelabels{$_},$count{H},$count{M},$count{L},$count{R},$count{F}\n";
	};
};
```