ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF HUMANITIES

DEPARTMENT OF LINGUISTICS

Multiple Pronunciations Modeling of Speaker Independent, Continuous Speech Recognition for Afaan Oromoo

A Thesis Submitted to School of Graduate Studies of Addis Ababa University in Partial Fulfillment for the Requirement of Masters of Science in Computational Linguistics

By

ONESMOS AMBERAS

January, 2013

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF HUMANITIES

Multiple Pronunciations Modeling of Speaker Independent, Continuous Speech Recognition for Afaan Oromoo

By

ONESMOS AMBERAS

Signature of the Board of Examiners for Approval

Name	Title	Signature	Date
Dr. Sebsibe H/Mariam	Advisor		
Dr. Wondewosen Tesfaye	Advisor		17/04/2013
Dr. Solomon Tefera	Examiner		17/04/2013
Ato Feda Negesse	Examiner		April 17, 2013

DEDICATION

This thesis is dedicated to my mother, Shumate Barkesa.

Acknowledgement

Glory is to the Almighty God that I came to this point in life. Next, I wish to express my deepest gratitude to my advisors, Dr. Sebsibe H/Mariam and Dr. Wondewosen Tesfaye. Their guidance, inspiration and advice throughout my study finally led me to successfully complete my thesis.

Next, I would like to show my sincere gratitude to my parent, brothers and sisters especially Josi A. who has been with me from the beginning with great care and warm encouragement.

Finally, I am deeply indebted to my friends Dire Girma and Tsegaye Hayilu who have been with me in all aspects from the early beginning to the end.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Appendices

List of Abbreviations

ASR:	Automatic Speech Recognition
HMM:	Hidden Markov Mode
HTK:	Hidden Markov Model Toolkit
LPC:	Linear Predictive Coding
MFCC:	Mel Frequency Cepstral Coefficient
LPCC:	Linear Predictive Cepstral coefficient
PLP:	Perceptual Linear Prediction
OCR:	Optical Character Recognition
LG:	Language Model
OOV:	Out of Vocabulary
MP:	Multiple Pronunciations
SCHMM	Semi-continuous Hidden Markov Model
VQ	Vector Quantization
CUED	Cambridge University of Engineering Department
SMT	Statistical machine translation

Abstract

Automatic speech recognition, especially speaker independent continuous speech recognition, is characterized by great variability in word pronunciation, including many variants that differ grossly from dictionary prototypes. This is one factor in the poor performance of automatic speech recognizers on speaker independent speech recognition.

One approach to handling this variation consists of expanding the dictionary with phonetic substitution, insertion, and deletion rules. This study aimed at modeling multiple pronunciations in speaker independent continuous speech recognition for Afaan Oromoo to handle pronunciation variation. Hidden Markov Model and the Hidden Markov Modeling Toolkit were used to implement it. For developing model for the language under study, a corpus containing 754 sentences collected from Bariisaa news paper and Afaan Oromoo bible (New Testament) was used. The data collected was preprocessed in line with the requirements of HTK. Phonemes were taken as base unit for recognition. Knowledge based pronunciation variation modeling technique was used for modeling words with multiple pronunciations. Thus two models were developed; one with canonical pronunciation and the other with alternate pronunciation to compare their relative performance.

Accordingly, the performance achieved using canonical pronunciation was 81.09% and 83.82 % correct for sentences and words respectively with word accuracy of 80.91% while the performance of alternate pronunciation was 83.08% and 85.11% sentences and words correct respectively with 82.52% word accuracy.

CHAPTER ONE

1.1. Introduction

For human being, speech constitutes a very efficient means of communication. This has induced many people to think that speech might also be a very efficient means of communication between human being and machine (Kessen, 2002). For this reason, attempts have been made to use speech as an input to computer. The term automatic speech recognition (ASR) is used for the technology that is required to transform speech in to text (Hämäläinen, 2009). In automatic speech recognition (ASR) systems acoustic information is sampled as a signal suitable for processing by computers and fed into a recognition process. The output of the system is a hypothesis for a transcription of the utterance.

Since the emergence of the first ASR, substantial progress has been made in the field of ASR. What started as recognition of a given digit spoken in isolation by a single speaker has now evolved to speaker independent, continuous large vocabulary recognition system (Kesarkar, 2003). In spite of the progress that has been made, a gap still exists between the performance of human being and machine on speech recognition.

There are a number of differences in the way speech is decoded by human being and by machine that could explain why ASR performance has not yet reached the same level of performance as human being speech recognition. One of the main differences between human and machine speech recognition is that human being use much more information for speech decoding than machines do. For instance, most human beings use two ears for hearing, where as speech recognizer usually process a single stream of speech. Furthermore, a speech recognizer can only

recognize the words that are contained only in its vocabulary. Another difference is that human beings have certain expectations on the kind of speech that is likely to be produced. Other information that machine can use only to a limited extent compared to human beings is information on intonation, stress, speaking rate and pronunciation variation (Kessens, 2002).

Words are pronounced differently by different people, a phenomenon called "pronunciation variation (Crèmelie, 1999). Pronunciation variation refers to the phenomena of words never being pronounced in exactly the same way by different people (between speaker variations) or even by the same (within speaker variation). Between speakers variation is caused by differences in vocal tract geometry, age, gender, regional and social accent (dialect) while within speaker variation is caused by speaking style, speaking rate, emotional state of the speaker , allophonic variation and super segmental features (Hämäläinen, 2009). Multiple pronunciation modeling is handling variation that may occur between the orthography and the actual utterance through adding variants to a lexicon (Kessens, 2002).

1.2. Statement of the Problem

Automatic speech recognition (ASR) has made great strides with the development of digital signal processing. But despite of all these advances, machines cannot match the performance of their human counterparts in terms of accuracy, especially in case of speaker independent, continuous speech recognition (Kesarkar, 2003). In line with this, one constraint of the problem that attracts researcher for further investigation is how to handle multiple pronunciations in ASR system that may alleviate ASR performance.

When words are pronounced in the same way, automatic speech recognition (ASR) would be relatively easy to recognize (Seman, 2008). However, for various reasons words are almost always pronounced differently. This variation in pronunciation is a major problem in ASR because it alleviates performance of the ASR system. Since the presence of variation in pronunciation may cause errors in ASR, modeling multiple pronunciations is seen as a possible way of handling this variation (Wester, 2002).

In Afaan Oromoo words are pronounced in more than one ways, they usually have multiple pronunciations (MP). This can cause the performance of automatic speech recognizers to deteriorate if it is not well accounted for. A common approach to handle this problem is to use multiple pronunciations modeling; where alternate pronunciations are added to lexeme in a lexicon in order to fit the acoustic data better (Lyu, Chiang and Hsu, 2005).

Therefore, this study aims at multiple pronunciations modeling in continuous speaker independent for Afaan Oromoo ASR system in order to handle pronunciation variations.

1.3. Objective of the Study

1.3.1. General Objective

The general objective of the study is to handle multiple pronunciations in Afaan Oromoo ASR system in order to assess if multiple pronunciations have impact on ASR performance.

Accordingly, this research opts to model multiple pronunciations that exists within the same dialect technically known as idiolect of inter-speaker. It does not consider different accent across regional dialect of Afaan Oromoo. This is due to unavailable already prepared corpus and limitation of time for the study. Therefore, the study is confined to only Mecha dialect, specifically Wollega as corpus preparation along with considering the five Afaan Oromoo dialects is very tedious and time consuming.

CHAPTER TWO

Literature Review and Theoretical Frame Work

This part attempts to present related works done on the area of pronunciation variation and speech recognition for local and non local language and in addition, theoretical concepts of speech production system and automatic speech recognition.

2.1. Literature Reviews

In this section related works done for local and non local language on multiple pronunciation modeling, speech recognition along with tools and techniques implemented was overviewed.
Many researchers have undergone investigation in designing and developing the possibility of speech recognition for different languages and besides, multiple pronunciations modeling to alleviate poor performance of ASR system. We can broadly classify these investigations along two directions. The first one is researches done on foreign (non local) language and the 2nd on local language.

Accordingly, to support this statement with justification for non local language a research was conducted by (Wester, 2002) entitled: Pronunciation Variation Modeling for Dutch Automatic Speech Recognition. In order to achieve the objective, a general procedure for modeling pronunciation variation was proposed. This procedure affects all three levels of the CSR at which modeling can take place: i.e. the lexicon, the phone models and the language model. This means that variants were added to the lexicon and language models, and that the phone models were retrained on a retranscription of the training material obtained through forced alignment. The

result obtained was 12.75%. WER for base line system. Adding pronunciation variants to the lexicon leads to an improvement of 0.31% compared to the baseline.

When, in addition, retrained phone models are used, a further improvement of 0.22% was found, and finally, incorporating variants into the language model led to a further improvement of 0.15%. Totally, significant improvement of 0.68% was found for modeling within-word

Totally work done in this paper was promising to handle problems related to pronunciation variation in ASR system.

When we come to the local language, certain researches were conducted for Amharic, Afaan Oromoo and Tigrinya languages regarding ASR but a few works have been done on pronunciation variation modeling. Solomon (2008) has conducted research on pronunciation variations of Amharic language. The study regarded the five Amharic dialects: Addis Ababa, Gojam, Gondor, wollo and Menz. HMM and HTK were used for experimentation.

The study used a corpus of 172 sentences selected from two sources: Woy Addis Ababa and Mekoya. Phones were used as unit base for the recognizer. The performance obtained was 52 % word accuracy for a model with multiple pronunciations and 47% for canonical.

The finding proved that it is possible to use multiple pronunciation dictionaries as a basic unit of phone based speech recognition for Amharic multiple pronunciation modeling to handle pronunciation variation.

When we come to the target language, Afaan Oromoo, there is no research done on this language by graduate students regarding multiple pronunciations modeling but on ASR.

Ashenafi (2009) has conducted research entitled 'A Speech Recognition System for Afaan Oromoo' using Hidden Markov Model and Sphinx. In the progression, a speech corpus was

constructed using 50 Afaan Oromoo words read by 20 persons. The experimentation involves construction of context independent and context dependant. Accordingly, 82.83% and 81.081% word level accuracy was obtained for context dependant phoneme based model and context independent phoneme based model respectively.

Similar work was done by Kasehun (2010) entitled A Continuous, Speaker Independent Speech Recognizer for Afaan Oromoo. 70 phrase databases recorded by 30 speakers of which 2/3 used for training and 1/3 for testing. The data were recorded using praat which is open source software. He used HMM and Sphinx system. In this research the evaluation performance depicts recognizer performance 68.514% with sentence accuracy of 28% for context independent and phoneme based trigram performance of 89.459% with sentence accuracy of 42%.

2.2. Theoretical Frame Work

This part attempts to overview theoretical concepts of speech production system and ASR.

2.2.1. Human Speech Production System

It must be said that speech does not start in the lungs. It starts in the brain. After the creation of the message and the lexico-grammatical structure in our mind, we need a representation of the sound sequence and a number of commands which will be executed by our speech organs to produce the utterance. After this mental operation we come to the physical production of sounds. Speech, then, is produced by an air stream from the lungs, which goes through the trachea and the oral and nasal cavities (Trujillo, 2003).

In order for communication to take place, a speaker must produce a speech signal in the form of a sound pressure wave form that travels from the speaker's mouth to a listener ears. Although the majority of the pressure wave originates from the mouth, sound also emanates from the nostrils, throats and cheeks. Speech signals are composed of a sequence of sound that serves as a symbolic representation for a thought that the speaker wishes to relay to the listener (Deller 2000).

The speech wave form is an acoustic sound pressure wave that originates from voluntary movement of anatomical structure which makes up the human speech production system (Tatham and Morton, 2006). When we speak, air comes out through the lungs and it is interfered at various places of speech organ for the production of sounds.

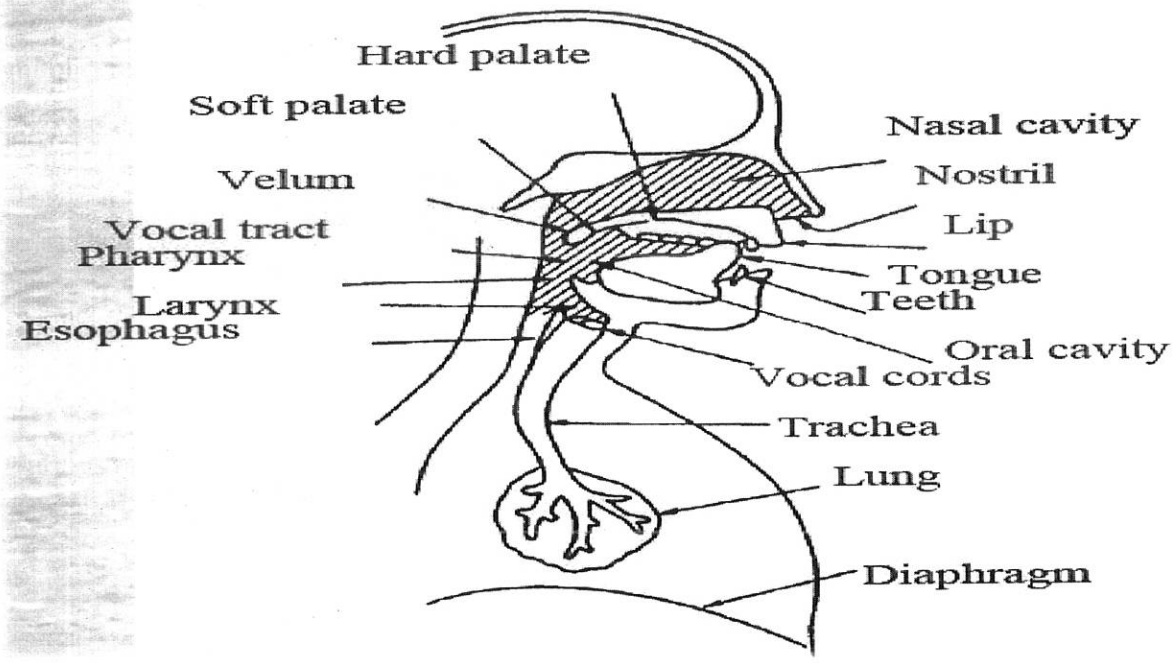

Fig.2.1. Human speech production mechanism (extracted from Deller, 2000)

Producing different speech sounds depends on the movement of speech organs. It is essential to know the movement and the placement of each organ to produce particular sounds (Honda, 2003).

Speech production with the help of speech organ is produced in the following ways (cited by Kassehun, 2010)

- Lungs provide energy = respiration
- Vocal folds convert the energy in to audible sound = phonation
- Articulators transform the sound in to intelligible speech = articulation.

A. Respiration

In normal speech, the action of the respiratory apparatus during exhalation provides a continuous stream of air with sufficient volume and pressure to initiate phonation. The stream of air is modified in its course from the lungs by the facial and oral structures giving rise to the sound symbols that are recognized as speech.

B. Phonation

Phonation is the process by which energy from the lungs in the form of air pressure is converted into audible vibrations (Trujillo, 2003). The phonatory process, or voicing, occurs when air is expelled from the lungs through the glottis, creating a pressure drop across the larynx. When this drop becomes sufficiently large, the vocal folds start to oscillate. The phonation process occurs at the larynx. The larynx has two horizontal folds of tissue in the passage of air; they are the vocal folds. The gap between these folds is called the glottis. The glottis can be closed, then, no air can pass. Or it can have a narrow opening which can make the vocal folds vibrate producing the

"voiced sounds". Finally, it can be wide open, as in normal breathing, and, thus, the vibration of the vocal folds is reduced, producing the "voiceless sounds". After it has gone through the larynx and the pharynx, the air can go into the nasal or the oral cavity. The velum is the part responsible for that selection. Through the oral-nasal process we can differentiate between the nasal consonants and other sounds (Steven, 1998).

C. Articulation

Finally, the articulation process is the most obvious one: it takes place in the mouth and it is the process through which we can differentiate most speech sounds. In the mouth we can distinguish between the oral cavity, which acts as a resonator, and the articulators, which can be active or passive: upper and lower lips, upper and lower teeth, tongue (tip, blade, front, back) and roof of the mouth (alveolar ridge, palate and velum). So, speech sounds are distinguished from one another in terms of the place where and the manner how they are articulated (Trujillo, 2003).

2.2.2. Automatic Speech Recognition (ASR)

Designing a machine that mimics human behavior, particularly the capability of speaking naturally and responding properly to spoken language, has intrigued engineers and scientists for centuries (*Juangand Rabiner, 2004*). A field that aims at the development of technologies that enable human beings to talk naturally with computers and or other devices is called spoken language processing. It refers to technologies related to speech recognition (converts speech into words), text-to-speech (converts text to speech), and speech understanding. (Solomon, Martha and Wolfgang, 2009).

2.2.2.1. Types of ASR

Speech recognition system can be classified on the basis of the constraints under which they are developed and which they consequently impose on their users. These constraints include: speaker dependence, type of utterance, size of vocabulary, linguistic constraints, types of speech and environment (cited by Solomon, 2005).

A. Types of Speaker

In this there are two constraints: speaker dependant and speaker independent. A speaker dependant speech recognition system requires the users to be involved in its development. It operates for speaker/s involved in the training. On the other hand, Speaker independent system is capable of recognizing speech from people whose speech the system has never been exposed to before (Jurafsky 2009). This system is difficult to develop because the application needs to recognize large, heterogeneous population of speakers however it is flexible and natural.

B. Vocabulary Size

The number of words in the vocabulary is a constraint that makes a speech recognition system small, medium or large. Small vocabulary system is that which has a vocabulary size in range of 1-99 words. Medium 100-999 words while large vocabulary comprises 1000 and above (Deller, Proakis and Hansen 1993)

C. Types of Speech

A speech recognizer can be developed to recognize only read speech (discrete) or to allow the user speak spontaneously (continuous). Only read speech (Isolated) word recognition, in which

each word is surrounded by some sort of pause, is much easier than recognizing continuous speech, in which words run into each other and have to be segmented. Speech is said to be continuous when it is uttered as a continuous flow of sounds with no inherent separation between them (Jurafsky, 2009).

D. Linguistics Constraints

Most of the present speech recognition systems are unable to reliably determine the identity of a speech input (phone or word) based on the speech signal alone. To improve reliability, linguistic constraints are put on a recognizer by using a language model and pronunciation dictionary. They capture syntactical and lexical constraints respectively. The more constrained the rule of a language in the recognizer, the less freedom of expression the user has in constructing spoken message.

E. Environment

Speech recognizer may require the speech to be clean from environmental noises, acoustic distortion and transmission channel distortion or they may ideally handle any of these problems. Current speech recognizers give better performance in carefully controlled environment. Their performance rapidly degrades when they are applied in noisy environment. This is due to that the noise can take the form of speech from other speakers, equipment sound, air condition, factory and from the speaker himself in the form of lips smacks, breathe takes, cough or sneezes (Ashenafi, 2009).

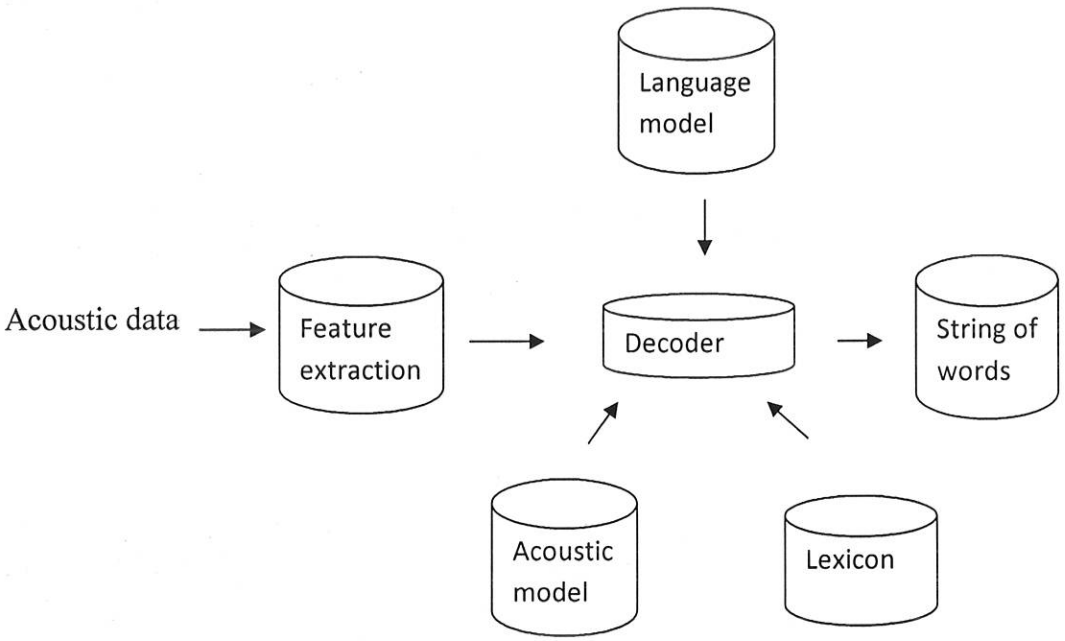

Fig.2.2. Basic Components of Speech Recognizer System (Holmes and Holme, 2001).

2.2.2.2. Approaches of ASR

From the early days of speech technology, automatic speech recognition system has been using various approaches in order to achieve better performance. Accordingly, major approaches used in automatic speech recognition are the followings (Anusuya, 2009).

A. Acoustic Phonetic Approach

The earliest approaches to speech recognition were based on finding speech sounds and providing appropriate labels to these sounds. This is the basis of the acoustic phonetic approach (Hemdal and Hughes 1967), which postulates that there exist finite, distinctive phonetic units (phonemes) in spoken language and that these units are broadly characterized by a set of acoustics properties that are manifested in the speech signal over time.

Even though, the acoustic properties of phonetic units are highly variable, both with speakers and with neighboring sounds (the so-called co articulation effect), it is assumed in the acoustic-phonetic approach that the rules governing the variability are straightforward and can be readily learned by a machine. The first step in the acoustic phonetic approach is a spectral analysis of the speech combined with a feature detection that converts the spectral measurements to a set of features that describe the broad acoustic properties of the different phonetic units.

The next step is a segmentation and labeling phase in which the speech signal is segmented into stable acoustic regions, followed by attaching one or more phonetic labels to each segmented region, resulting in a phoneme lattice characterization of the speech.

The last step in this approach attempts to determine a valid word (or string of words) from the phonetic label sequences produced by the segmentation to labeling. In the validation process, linguistic constraints on the task (i.e., the vocabulary, the syntax, and other semantic rules) are invoked in order to access the lexicon for word decoding based on the phoneme lattice. The acoustic phonetic approach has not been widely used in most commercial applications.

B. Pattern Recognition Approach

This approach basically uses speech pattern directly without explicit feature determination (in the acoustic phonetic sense) and segmentation (cited by Ashenafi 2009). The pattern-matching approach (Rabiner and Juang, 1993) involves two essential steps namely, pattern training and recognition of pattern via comparison. The essential feature of this approach is that it uses a well formulated mathematical framework and establishes consistent speech pattern representations, for reliable pattern comparison, from a set of labeled training samples via a formal training

algorithm. A speech pattern representation can be in the form of a speech template or a statistical model (e.g., HMM) and can be applied to a sound (smaller than a word), a word, or a phrase.

C. Stochastic Approach:

Stochastic modeling entails the use of probabilistic models to deal with uncertain or incomplete information (Anusuya, 2009). In speech recognition, uncertainty and incompleteness arise from many sources; for example, confusable sounds, speaker variability, contextual effects, and homophones words. Thus, stochastic models are particularly suitable approach to speech recognition. The most popular stochastic approach today is hidden Markov modeling (Rabiner, 1988). A Hidden Markov Model is characterized by a finite state Markov model and a set of output distributions. The transition parameters in the Markov chain models, temporal variability, while the parameters in the output distribution model spectral variability. These two types of variability are the essence of speech recognition.

The HMM, being a probability measure, is amenable for incorporation in a larger speech decoding framework which included a language model. The use of a finite-state grammar in large vocabulary continuous speech recognition represented a consistent extension of the Markov chain that the HMM utilize to account for the structure of the language Since this approach avoids a direct comparison with stored templates, generally exhibits higher recognition performance and is currently applied in most automatic speech recognition system. The pattern recognition has four stages (Juang and Rabiner, 1993):

Feature extraction, in which the important features are extracted from the input signal and represent it a form of feature pattern. Feature extraction includes LPC, LPCC and MFCC (Kesarkar, 2003).

Pattern training: in which one or more test patterns are corresponding to speech sounds of the same class are used to create a pattern representative of the feature of that class (Rabiner 1993). The resulting pattern, generally called reference pattern, can be template, derived from some type of averaging technique, or a model that characterizes the statistics of the feature of the reference pattern.

Pattern classification: in which the unknown test pattern is compared with each class reference patterns and a measure of similarity between the test pattern and each reference pattern is computed.

Decision logic: this is the step in which the reference pattern similarity scores are used to decide which reference pattern best matches the unknown test pattern.

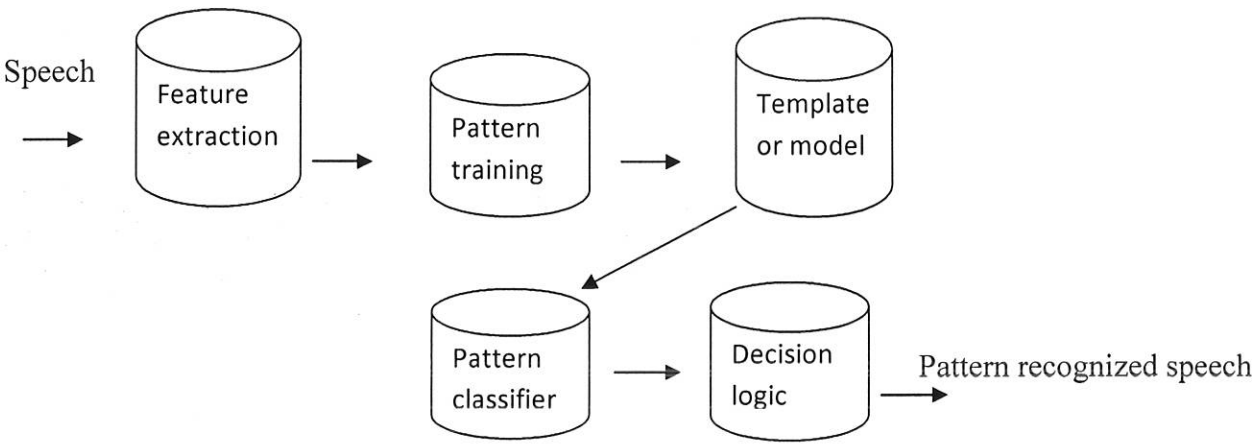

Figure: 2.3. Pattern Recognition Speech Recognizer (Rabiner and Juang, 1993)

D. Artificial Intelligence Approach (Knowledge Based Approach)

The Artificial Intelligence approach is a hybrid of the acoustic phonetic approach and pattern recognition approach. In this, it exploits the ideas and concepts of Acoustic phonetic and pattern recognition methods. Knowledge based approach uses the information regarding linguistic, phonetic and spectrogram (Holmes and Holmes, 2001). Some speech researchers developed recognition system that used acoustic phonetic knowledge to develop classification rules for speech sounds. While template based approaches have been very effective in the design of a variety of speech recognition systems; they provided little insight about human speech processing, thereby making error analysis and knowledge-based system enhancement difficult (Anusuya, 2009).

On the other hand, a large body of linguistic and phonetic literature provided insights and understanding to human speech processing. In its pure form, knowledge engineering design involves the direct and explicit incorporation of expert speech knowledge into a recognition system. This knowledge is usually derived from careful study of spectrograms and is incorporated using rules or procedures. However, this approach had only limited success, largely due to the difficulty in quantifying expert knowledge. Another difficult problem is the integration of many levels of human knowledge phonetics, phonotactics, lexical access, syntax, semantics and pragmatics. Alternatively, combining independent and asynchronous knowledge sources optimally remains an unsolved problem (R. Rabiner, 1989).

In more indirect forms, knowledge has also been used to guide the design of the models and algorithms of other techniques such as template matching and stochastic modeling. This form of

knowledge application makes an important distinction between knowledge and algorithms. Algorithms enable us to solve problems. Knowledge enables the algorithms to work better. This form of knowledge based system enhancement has contributed considerably to the design of all successful strategies reported. It plays an important role in the selection of a suitable input representation, the definition of units of speech, or the design of the recognition algorithm itself.

2.2.2.3. Fundamentals of Speech Recognition

The underlying assumption behind any recognition system is that the waveform of a speech signal that comes out of a speaker's vocal apparatus is a realization of the concept that was in the form of symbols in his/her mind. When a source conceives an idea to speak out, it was understood symbolically. The moment it gets out to the channel, it materializes in the form of speech signals or sound waves. Thus, one direct and possible approach for a computer based speech recognition system to recognize an utterance is inferring the original symbols from the speech signals (Young et.al, 2002).

To effect this reverse operation of recognizing the underlying symbol sequence given a spoken utterance, first the speech waveform should be digitized and important features must be extracted from the digitized format. Then the extracted features out of the continuous speech waveform should be converted to a sequence of equally spaced discrete parameter vectors

Speech recognizer can be said to possess three components (cited by Zegaye, 2003): spectral analysis, language modeling and acoustic modeling and decoding.

1. Spectral analysis: Front End Processing

Front end analysis refers to the first stage of ASR, whereby the input acoustic signal is converted to a sequence of acoustic feature vector (Cheng and Abdulla, 2005). Ideally the method of front end analysis should preserve all the perceptually important information for making phonetic distinction, while not being sensitive to acoustic variations that are irrelevant phonetically.

The main task of the front-end is to extract features from a speech signal. The aim is to sufficiently represent the characteristics of the speech signal with reduced redundancy. Features are extracted based on frames (windows) (Kesarkar, 2003).

There are three major types of feature extraction front-ends that are commonly used in speech recognizers (Cheng and Abdulla, 2005). They are Linear Prediction Cepstral Coefficients (LPCC), Perceptual Linear Prediction Coefficients (PLP) and Mel Frequency Cepstral Coefficients (MFCC).

Linear Prediction Cepstral Coefficient (LPCC)

Linear Prediction Cepstral Coefficients (LPCC) has been commonly used in many speech recognition applications for many years. The notion behind LPCC is to model the human vocal tract by a digital all-pole filter and passes through the following (Mantha, Duncan and Zhao, 2001).

A. Pre-emphasis and Windowing

The first step of the algorithm is pre-emphasis. The idea of pre-emphasis is to spectrally flatten the speech signal and equalize the inherent spectral tilt in speech. Pre-emphasis is implemented by a first order FIR digital filter.

B. Linear Predictive Analysis

In human speech production, the shape of the vocal tract governs the nature of the sound being produced. In order to study the properties quantitatively, the vocal tract is modeled by a digital all-pole filter. The LPC lies on the assumption that the space between the vocal cords called glottis, produces speech signal which is characterized by its intensity (loudness) and frequency, which determines the speech of the sound. LPC analyzes the speech signal frames by estimating the formants, removing their effects from the speech signal intensity and frequency of the remaining buzz (Wiggers, 2001).

The basic intention of LPC is to determine the formants from the speech signal which is done by different equations called a linear predictor.

C. Cepstral Analysis

Cepstral analysis refers to the process of finding the cepstrum of a speech sequence. Cepstrum, whose spelling is formed by shuffling the characters of the word spectrum, is a time-domain representation of a signal. Cepstrum is defined as the inverse Fourier transform of the logarithm of a signal's spectrum. It has also been used to determine the fundamental frequency of human speech (Kesarkar, 2003).

Perceptual Linear Prediction Coefficients (PLP)

Perceptual Linear Prediction (PLP) coefficient is another feature extraction technique, which tries to emulate the human auditory system.

Mel Frequency Cepstral Coefficient

Mel Frequency Cepstral Coefficients (MFCC) is one of the most commonly used feature extraction front-ends in speech recognition systems (Kesarkar, 2003). It deals with power spectrum of speech signal which describe the frequency content of the signal over time. The purpose is to reduce the number of data characterizing the signal and shows a limited parameter or coefficient discriminating and robust. The technique is so-called FFT-based, which means that feature vectors are extracted from the frequency spectra of the windowed speech frames.

2. Language Model

Language modeling is a task assigning a probability to a given sequence of words. It is crucial and indispensable for many speech and natural language applications such as automatic speech recognition (ASR), statistical machine translation (SMT) and optical character recognition (OCR) (Kim, 2004). It helps the applications especially in two ways. First, it reduces the search space of the problems. Most speech and natural language processing problems can be regarded as finding the most likely answer (word strings) given an input data (test set), and the main difficulty lies in that innumerable candidate answers are possible which implies that it is impossible to search all possible candidate answers in most practical problems. Provided by some probabilities estimated from a language model (LM), the search space can be effectively reduced by ignoring unlikely candidates, and thus the search problem becomes feasible. Second, language model actually improves an application's performance by providing contextual information. For instance, in OCR it is sometimes difficult to distinguish the letter '**l**' and the number'**1**'. In ASR also, it is practically impossible to distinguish **cent**, **sent**, and **scent** from one another unless some contextual information is given.

A speaker independent continuous speech recognition system is dependent on linguistic knowledge. Hence, incorporation of linguistic knowledge of the language in the form of language model, p (w) is very essential in continuous speech recognition system (Zegeye, 2003). N-gram is the most popular language model (LM) used to illustrate how the LM assigns probability to given input word strings (Kim, 2004).

The ASR problem is to find the most likely word string W from the given acoustic evidence (input data)

$\grave{W}$= argmax $\frac{P(W/A)}{W}$

By applying Bayes' formula equation can be rewritten as

P (W|A) = P (W)P(A|W)/ P(A) and since A is fixed—acoustic evidence is already given and doesn't change over the recognition process. ASR problem can be decomposed into two parts, the acoustic modeling problem, and the language modeling problem.

ˆW= argmax p/ (w)

P| (W) is Language Model

P| (A|W) is Acoustic Model

According to (Zegaye, 2003), small vocabulary recognition system do not rely on language model to accomplish their tasks because they are mainly used in command and control signal that the vocabulary has to respond. Such language model may incorporate syntactic or semantic constraints. When only syntactic constraint is used, the language model is called a grammar.

A large vocabulary speech recognition system, however is generally depends on linguistic knowledge. Hence, incorporation of knowledge of the language, in the form of a language model is essential for large vocabulary system.

3. Acoustic Model

Speech is in essence just a sequence of different sounds. Our brains are tuned to classify these sounds into basic phonetic units, or phonemes. From a sequence of phonemes we can distinguish words. From a pattern recognition point of view, this is quite an astonishing feat considering that the brain is also able to comprehend speech produced in different environments and by different speakers. Devising an algorithm for a computer to do the same is not a trivial matter (Mansikkaniemi, 2010).

An acoustic model is a file that contains statistical representations of each of the distinct sounds that makes up a word. Each of these statistical representations is assigned a label called phoneme (Jurafky, 2009). The core of an acoustic model lies in the capability of the feature vector to capture the distinctive property of the speech. Speech recognition engines require two types of files to recognize speech (Odelly 1995). They require an acoustic model, which is created by taking audio recordings of speech and their transcriptions (taken from a speech corpus), and compiling them into a statistical representations of the sounds that make up each word through a process called training. They also require a language model or grammar file, a file containing the probabilities of sequences of words.

Once the signal has been transformed in to a parameterized form, it must be recognized or decoded and turned in to the underlying sequence of symbols. This decoding process requires

patters against which unknown utterances can be compared. The acoustic model, P (O/W), provides the probability that the speech data was observed for a given word sequences.

The required probability distribution could be found by obtaining many examples of each word W and collecting the statistics of the corresponding vector sequences (Young, 1996). However, this is impractical for large vocabulary system instead word sequence is decomposed in to phonemes. Probability for word sequence is generated as a product of the acoustic and language model probability. The process of combining these two probability scores and sorting through all plausible hypotheses to select the one with maximum probability is called decoding or search (Ganapathiraju, 2002).

CHAPTER THREE

Multiple Pronunciations in Afaan Oromoo

3.1. Afaan Oromoo

From the broader perspective Afaan Oromoo is grouped under what is called the East Cushitic family which falls under the Afro-Asiatic super family (Kebede, 2009). Oromoo people are one of the major linguistic groups in Ethiopia. They live over a large area stretching from close to the Sudan border in the West, through Addis Ababa, and beyond Harar in the East, from Northern Kenya in the South, up and East of the rift valley, and to Wallo in the North" (Gragg, 1982). According to the 2007 statistics census agency the Oromoo people make up 27 million out of the 80 million of Ethiopia population.

Afaan Oromoo is one of the languages of the Lowland East Cushitic within the Cushitic family of the Afro-Asiatic Phylum (Bender, 1976, Gragg, 1982). According to Gadaa (1988) and Mahdi (1995), Afaan Oromoo is the 4th most widely spoken language in Africa after Swahili, Arabic and Hausa. The writing system used in Afaan Oromoo is the Latin based orthography, qubee. It officially replaced the former Ethiopic writing style in 1991after the fall down of Dergue reign (Tilahun, 1993). There are 32 letters in Afaan Oromoo of which 27 consonant including /‘/ and 5 vowels (Wikipedia). Afaan Oromoo is read in the way it is written. The fact that words are written and read in the way they sound would mean that word spellings or word pronunciations need not be memorized. This is the case because phonetic transcription is a rule-based technique for writing and reading unambiguously.

A a	B b	C c	CH ch	D d	DH dh	E e	F f	G g	H h	I i
[a]	[b]	[ʧ]	[ʧ]	[d]	[ɗ]	[e]	[f]	[g]	[h]	[i]
J j	K k	L l	M m	N n	NY ny	O o	P p	PH ph	Q q	R r
[ʤ]	[k]	[l]	[m]	[n]	[ɲ]	[o]	[p]	[p']	[k']	[r]
S s	SH sh	T t	U u	V v	W w	X x	Y y	Z z		
[s]	[ʃ]	[t]	[u]	[v]	[w]	[t']	[j]	[z]		

Afaan Oromoo Alphabet (extracted from Wikipedia)

Multiple pronunciations is different ways in which a given word is pronounced by different or the same individual (Seman, 2008). This phenomenon is also common in Afaan Oromoo. Words are pronounced differently by the same speaker or a group of speakers in Afaan Oromoo which refers to intra-speaker and inter-speaker respectively. This phenomenon is one issue to be considered in ASR system because unlike human being machine cannot tolerate variation; thus unable to recognize.

Multiple pronunciations (pronunciation variation) can be divided in to two main kinds: these are variation within-word and cross-word (Kessens, 2002). The first one is type of pronunciation variation that becomes apparent in a careful phonetic transcription of speech, in the form of insertions, deletions or substitutions of phones relative to a single, normative (canonical) transcription of the words. It is variation in the order and number of phones a word consists. The second one is type of pronunciation variation happens in continuous speech recognition system which occurs over word boundaries due to reduction, contraction and cliticization. It is variation in the acoustic realization of phones. This research is interested in the first kind of variation because this variation is expected to be more detrimental to speech recognition than the second one.

Pronunciation variation can be intra-speaker variability or inter-speaker variability (Wester, 2002).

1. Intra-speaker

Intra-speaker is variation in pronunciation for one and the same speaker. There are numerous factors that influence the degree of intra-speaker pronunciation variation that is encountered in speech. These include: speaking styles, speaking rate, supra segmental features, coarticulation and emotional state of the speaker (Kessens, 2002). For example, in Afaan Oromoo /m/ is pronounced as either /m/ or /n/ when followed by /b/ like in the word simbirroo (bird), simboo (beauty). In Afaan Oromoo /h/ can be omitted when at the beginning of words resulting slight difference in pronunciation.

Example, hoolaa ⟶ oolaa

Ho'a ⟶ o'a

/h/ is some time used instead of /'/ like in the word 'ta'a'..... taha resulting difference in pronunciation. Afaan Oromo permits, metathesis, i e, transposing the order or position of C_1C_2 in the language which has contribution for variation in pronunciation (Tilahun, 1998).

Example, a<u>r</u>faasaa (autumn) ⟶ a<u>f</u>raasaa

Dhoksuu (to hide) ⟶ dhoskuu

arfaffaa (4th) ⟶ afraffaa

4. **Inter-speaker variability**

This is variation among different speakers. The variation can be due to factors such as vocal tract differences, age, gender, regional accent, dialect and voice quality (Kessens, 2002).

Accent refers to the way in which a speaker pronounces, and therefore refers to a variety which is phonetically and.or phonologically different from other varieties (Chambers and Peter, 2004). This is prominent in Afaan Oromoo which results pronunciation variation. Example, miila or miilla (leg) jaallata or jaalata (he loves).

Dialect refers to a distinct variety of a language, especially one spoken in a specific part of a country or other geographical area. The criterion for distinguishing dialect from language is taken, in principle, to be that of mutual intelligibility. A dialect develops in time through linguistic changes in different regions where it was spoken (Kebede, 2008: 11). There are five Afaan Oromoo dialects spoken in Ethiopia, according to Gragg (1976) and Kebede (1998). These are: Maccaa (mecha), Hararge, Tulema, Borena and Arsi-Bale dialects. People from different dialect of this language usually pronounce a given word differently.

For example, 'eessa'(where) is read as 'ee ss a' in Mecha (Wollega) and is read as 'ee ch a' in Tulema (Showa). In the same way in word of belongness like keeti (yours), keenya (ours) /k/ is read as /t/ in , Borana and Arsi- Bale dialect while it is read in usual way in other dialects. These sources of variation all contribute to the fact that a word is never pronounced in exactly the same way by the same or different speakers (Strik and Cucchiarini, 1999).

The objective of automatic speech recognition (ASR) is to recognize what a person has said, i.e., to derive the string of spoken words from an acoustic signal. Due to the above described variation this objective becomes more difficult to be achieved, as the pronunciation variation may lead to recognition errors. A common approach to solving this problem is to use pronunciation modeling; where multiple pronunciations are added to each lexeme in a lexicon in

order to fit the acoustic data better (Hämaläinen, 2009). When dialogues between humans and computers are more natural, the ASR handles more conversational speech. Therefore, avenues are sought to model pronunciation variation in order to handle this variation.

3.2. Approaches of Pronunciation variation Modeling

There are two methods for pronunciation variation modeling.

3.2.1. Knowledge Based Pronunciation Variation Modeling

In this approach information about pronunciation is derived from knowledge sources, such as pronunciation dictionaries, phonological rules hand-crafted by linguistic experts or extracted from the literature (Amdal and Eric Fosler - Lussier, 2002). This approach makes use of existing knowledge that was derived by experts. This can be dictionaries or results from linguistic studies on pronunciation variation. The gathered information is often used to derive rules that are able to generate typical pronunciation variants from canonical pronunciations or from the orthography of a word. The pronunciation variants that are generated by applying the rules can then be added to the dictionary.

For Afaan Oromoo there is no dictionary prepared with multiple pronunciations. Thus, the researcher used linguistics knowledge to develop alternate pronunciation dictionary.

The advantage of this approach is that it is completely task independent, since it uses general linguistic and phonetic rules and can thus be used across corpora and especially for new words that are introduced to the system.

The drawback however is that the rules are often very general and thus many variants are generated, some of which might not be observed very often. On the other hand the existing knowledge might not cover all aspects that are needed for the current task and thus not enough variants might be generated. Furthermore no information on how often the generated variants appear in the data under consideration is given (Wester, 2002).

3.3.2. Data Driven Pronunciation Model

Data-driven approaches try to derive the pronunciation variants directly from the speech signal. This can help avoiding over-generation since only variants that really occur in the data are used. It furthermore allows the computation of application likelihoods. On the other hand this method is very much database-dependent and variants that occur frequently in one speech corpus do not necessarily occur frequently in other corpora (Amdal and Eric Fosler - Lussier, *2002*).

3.3. Issues in Pronunciation Variation Modeling

One of the main challenges in pronunciation modeling is to know which variation we are attempting to model. The effects of the acoustic models, the lexicon, and the language model will interact, even the choices at the speech pre-processing stage will influence the variation modeling (Amdal and Fosler –Lessier, 2002).

According to (Wester, 2002) there are two questions which cover most of the issues that must be addressed when modeling pronunciation variation:

1. How is the information obtained that is required to describe pronunciation variation?
2. How is this information incorporated in the ASR system?

3.3.1. Obtaining Information

Information about pronunciation variation can be acquired from the data itself or through (prior) knowledge; also termed the data-derived and the knowledge-based approaches to modeling pronunciation variation. One can classify approaches in which information is derived from phonological or phonetics knowledge and/or linguistic literature under knowledge-based approaches (Seman, 2008). In contrast, data-derived approaches include methods in which manual transcriptions of the training data are employed to obtain information or automatic transcriptions are used as the starting point for generating lists of variants.

Although the above approaches are useful, to a certain extent, for generating variants, they all have their own drawbacks too. The linguistic literature, including pronunciation dictionaries are not exhaustive; not all processes that occur in spontaneous speech (or even read speech) are described in the linguistic literature, or are present in pronunciation dictionaries. Furthermore, a knowledge-based approach runs the risk of suffering from discrepancies between theoretical pronunciations and phonetic reality (Cucchiarini, 1993).

3.3.2. Incorporating the Information in ASR

After the pronunciation variants are obtained, the next question that must be addressed is how the information should be incorporated into the ASR system. There are different levels at which this problem can be addressed (Amdal and Fosler-Lussier, 2002).

- **Modeling Pronunciation at the Level of Lexicon**

With respect to the type of pronunciation variation to be modeled the choice is between variation within words and variation across word boundaries. This choice will be influenced by factors such as the type of ASR and the language which is used, and the level at which modeling will take place. Modeling within-word variation is an obvious choice if the ASR makes use of a lexicon with word entries, because in this case variants can simply be added to the lexicon (Strik and Cucchiarini, 1999).

The lexicon typically consists of the orthography of words that occur in the training and their corresponding phonetic transcriptions. During recognition, the phonetic transcriptions in the lexicon function as a constraint which defines the sequences of phonemes that are permitted to occur. The transcriptions can be obtained either manually or through grapheme-to-phoneme conversion. In pronunciation variation research one is usually confronted with two types of lexica: a canonical (or baseline) lexicon and a multiple pronunciation lexicon. A canonical lexicon contains the normative or standard transcriptions for the words; this is a single transcription per word. A multiple pronunciation lexicon contains more than one variant per word, for some or all of the words in the lexicon (Wester, 2002).

Modeling Pronunciation variation at this level needs adding of variants to the baseline recognition lexicon. In this way a lexicon is obtained that contains multiple pronunciations for some of the words. However, adding pronunciation variants to the lexicon usually also introduces new errors because the acoustic transcriptions of the added variants can be confused with those of other entries in the lexicon. This can be minimized by making an appropriate selection of the pronunciation variants (Amdal and Fosler –Lussier, 2002).

- **Modeling Pronunciation at the Level of Acoustic Models**

An obvious ways of optimizing the acoustic is using a procedure referred to as iterative transcribing. In this procedure, pronunciation variants are used both during training and recognition. The transcriptions and canonical lexicon are the starting point. These are used to train the first acoustic model. Subsequently, the pronunciation variants are added to the lexicon. For every word in the corpus for which pronunciation variants are present in the lexicon, the ASR selects the optimal one. In this way new updated transcriptions are obtained which in turn used to train new acoustic model. Updating the transcription and retraining the acoustic model can be repeated iteratively (Amdal and Fosler –Lessier, 2002).

- **Modeling Pronunciation at the Level of the Language Model**

In automatic speech recognition (ASR), language models assign weights to word sequences to discriminate between acoustically similar sequences. A new language model is calculated from the new automatic transcription of the training corpus.

CHAPTER FOUR

Methodology

4.1. Data Collection

The procedure of sentences selection from a text data base aims at both a phonetically rich and balanced collection of sentences with regard to the relative frequency of sub word units to be modeled. To accomplish phonetic richness, we should select sentences which contribute to the inclusion of all phonemes (Solomon 2005). Phonetic balance of the corpus is achieved through selecting those sentences which contribute to the preservation of the distribution of all phonemes of the language.

In contrast to other languages like English, there is no easily available electronic text source for Afaan Oromoo for this purpose. Therefore, two Afaan Oromoo texts were selected for corpus collection: Afaan Oromoo bible (New Testament) and Bariisaa news paper (2004 published). These texts were selected considering agendas that might be discussed in these texts that help us get phoneme coverage with optimal sentences. The corpus collected consist 754 sentences which are divided in to two parts: training data sets (553 sentences) and testing data set (201 sentences). The sentences were selected for training and testing randomly.

To build robust acoustic models, it is necessary to train them on a large set of sentences containing many words with phoneme comprehensive. Accordingly, to evaluate phoneme comprehensiveness a simple python program was used which displays all phonemes with their frequency. Phoneme /a/ has the highest frequency (1345 times) and /v/ the least (10 times); hence the corpus collected is phonetically balanced.

The corpus prepared was recorded using Sony Digital Recording version 3.2 and Toshiba lap top with CPU 2.26GHZ , RAM 2GB and hard disc capacity of 160GB with window 7 operating system. In addition, 48Hz frequency was used with hamming window.

As environment influences performance of ASR system, the recording was undertaken in a calm environment, at Western Wollega Dongoro Dissi Elementary School. Though the room was not sound proof, it is relatively silent and free from sounds like car and grinding mills.

Data	No of sentences	No of speakers	
		Male	Female
Training	553	5	5
Testing	201	2	2

Table 4.2: summery of the database

Besides, while selecting people for recording purpose, age and sex was considered. Accordingly, 10 people with varying age group from 18-34 of which 5 men and 5 women were used for training while 4 of which 2 male and 2 female used for testing purpose.

Age range	No of speakers
18-25	7
26-30	5
31-34	2

Table 4.3. Age variation of the individuals

Before starting to record, each individual was told how to read the sentences and if made mistake requested to reread. Every individual was made read almost equal number of sentences (55) for training and 50 sentences for testing respectively. Accordingly, the selected Afaan Oromoo sentences were recorded using Sony digital voice recorder.

4.2. Modeling Method

Hidden Markov model (HMM) is a statistical model, which has been used in diverse fields in which information occurs in sequences of discrete emissions. It is a powerful statistical tool for modeling generative sequences that can be characterized by an underlying process generating an observable sequence (Wester, 2002).

HMMs have found application in many areas interested in signal processing, and in particular speech processing.The model has a finite number of states, each of which has a distinctive frequency distribution over the 'alphabet' of possible emissions. The states are connected to one another by a set of probabilities. Two sets of probabilities are associated with each state: a transition probability, which gives the probability of taking the transition, and an output or

emission probability density function, which specifies the probability of emitting each output symbol (Rabiner 1988).

An HMM is trained for each recognition unit (e.g. phones) defined in the system. We note the state into which each emission can be placed, trying to maximize the score at all times. The state path that scores the highest represents a demarcation of the sequence into partitions of different emission frequencies.

Once the model file has been parsed, two 2D matrices are created. One is the transition matrix, and the other is the emission matrix. The transition matrix has dimensions to contain transition probabilities from each state to every other state in the model, including itself. Transitions disallowed by the model topology are set to zero. All values are stored as log-probabilities for arithmetic reasons (cited by Zegeye, 2003).

The emission matrix contains the emission probabilities for each phone within each state. These are also stored as log probabilities. In its discrete form, a hidden Markov process can be visualized as a generalization of the urn problem. A genie is in a room that is not visible to an observer. In this hidden room there are urns X1, X2, X3 ... each of which contains a known mix of balls, each ball labeled y1, y2, y3... The genie chooses an urn in that room and randomly draws a ball from that urn. It then puts the ball onto a conveyor belt, where the observer can observe the sequence of the balls but not the sequence of urns from which they were drawn. The genie has some procedure to choose urns; the choice of the urn for the n-th ball depends only upon a random number and the choice of the urn for the $(n - 1)$-th ball. The choice of urn does not directly depend on the urns chosen before this single previous urn; therefore, this is called a

markov process. The Markov process itself cannot be observed, and only the sequence of labeled balls can be observed, thus this arrangement is called a "hidden Markov process" (R. Rabiner, 1989).

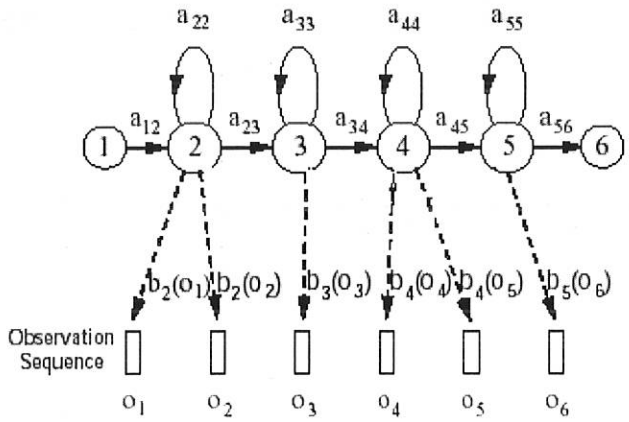

Fig.4.4. Hidden markov model chain (extracted from Rabiner 1988)

4.2.1. Elements of HMM

An HMM is characterized by the following (R. Rabiner 1989)

1. **N, the number of states in the model.**

Although the states are hidden, for many practical applications there is often some physical significance attached to the states or to sets of states of the model. Hence, for example, in the coin tossing, each state corresponds to a distinct biased coin. We can denote the individual state as $S = \{S_1, S_2 \ldots, S_N\}$ and the state at time t as qt.

2. **M, the number of distinct observation symbol per state**

The observation symbol corresponds to the physical output of the system being modeled. For example for coin tossed the observation symbols are head or tail. This can be denoted as $V = \{v_1, v_2 \ldots v_M\}$.

3. **The state transition probability distribution A = $\{a_{ij}\}$ where**

$a_{ij} = P[q_{t+1} = S_j / q_t = S_i],\ 1 < i, j < N$

4. **The observation symbol probability distribution in state j, B = {b, (k)},** where

$b_{j(k)} = P\,[v_k \text{ at } t / q_{t=Si}]\ \ 1 =< j =< N\ \ 1 =< k =< M.$

5. **The initial state distribution π =** $\{\pi_i\}$ where $\pi = p\,[q_1 = S_j],\ 1 =< I =< N$

Given appropriate value of N, M, A, B, and $\pi$the HMM can be used to give an observation sequence:

$O = o_1, o_2 \ldots o_T$ where each observation o_t is one of the symbol from v_t and T is the number of observation in the sequence.

Therefore, an HMM is specified by two scalars (*N* and *M*) and three probability distributions (*A*, B and π). (Huang, 1989)

4.2.2. HMM Topologies for Speech Recognition

Most topologies used in speech recognition are based on the assumption that there are three phases in the pronunciation of a phone (Wiggers, 2001). In the first phase the vocal tract is changing shape to pronounce the phone; this is called the on-glide of the phone. In this phase there may be some overlap with the preceding phone. In the second phase the sound of the phone is assumed to be pure and in the third phase the sound is released and the vocal tract starts to transit to the next phone. This is called the off-glide, some overlap with the next phone may occur here. This suggests that at least three states should be used in a phoneme based HMM.

In the three states, the first could represent the transition in to the phoneme, the second bears the steady state portion and the third depicts the transition out of the phoneme (Lee, 1987). Adding

more states means introducing more parameters and thus more degrees of freedom. Variations in a phoneme can be modeled more accurate but this also introduces a need for more training data to avoid under training. But a model should not be too large, a five state model does not work for phones that only occupy three time frames, so in larger models there should always be a 'short-cut' that can handle the shortest example in the training data (Wiggers, 2001).

4.2.3. The Three Basic Problem of HMM

There are three basic problems that must be solved for HMM model to be used in real world application (Stamp, 2012). These problems are the following.

1. Given the observation sequence $O = O_1\ O_2 \ldots O_T$ and model $\lambda = (A, B, \pi)$ how do we efficiently compute $P\ (O/\ \lambda)$, the probability of the observation sequence, given the model?

This problem can be tackled using forward-backward algorithm as follows (Stamp, 2012):

To calculate the probability of the observation sequence, $O = O_1\ O_2 \ldots O_T$, given the modelλ, i.e. $P\ (O/\ \lambda)$. The most straightforward way of doing this is through enumerating every possible state sequence of length T (the number of observation) $Q = q_1\ q_2 \ldots q_T$ Where q1 is the initial state.

The probability of the observation sequence O for the state sequence is $P\ (O/Q,\ \lambda)$.

Thus we get:

$$P\ (O/\ Q,\ \lambda) = b_{q1}(O_1),\ b_{q2}\ (O_2) \ldots b_{qT\ (}O_{T)}$$

This probability can be rewritten as

$$P\ (Q/\ \lambda) = \pi q1aq1q2aq2q3 \ldots bqT(OT)$$

The probability of such state can be

$$P\ (Q/\ \lambda) = \pi q1aq1q2aq2q3 \ldots aq_{t-1}q_T$$

The joint probability of O and Q i.e. the probability that O and Q occur simultaneously is the product of the above two terms

$P(O, Q/\lambda) = P(O/Q, \lambda) P(Q, \lambda)$

This probability depicts that initially we have t=1 and q1 with probability $\pi q1$ generating symbol O1 with probability $b_{q1}(O_T)$. As time changes to t+1 we make transition to state q2 with probability a_{q1q2} and generate symbol O_2 with probability $b_{q2}(O_2)$ this process continues.

2. Given the observation sequence $O = O_1\ O_2 \ldots O_T$, and modelλ, how do we choose a corresponding state sequence $Q = q_1\ q_2 \ldots q_t$, which is optimal in some meaningful sense?

   This problem can be solved through finding the optimal state sequence associated with the given observation sequence using viterbi algorithm.

3. How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O/\lambda)$?

   The Baum-Welch re-estimation algorithm can solve this problem.

4.2.4. Types of HMM

A. Continuous

Continuous mixture HMM models the acoustic observation directly using estimated continuous probability density functions without VQ, and has been shown to improve the recognition accuracy in comparison to the discrete HMM (R. Rabiner, 1989). In this approach quantization error can be eliminated by using a continuous density model instead of VQ codebook.

For speaker-independent speech recognition, mixture of a large number of probability density functions or a large number of states in single-mixture case are generally required to model characteristics of different speakers.

However, mixture of a large number of probability density functions will considerably increase not only the computational complexity, but also the number of free parameters that can be reliably estimated. In addition, the continuous mixture HMM has to be used with care as continuous probability density functions make more assumption than the discrete HMM, especially when the diagonal covariance Gaussian probability density is used for simplicity. To obtain better recognition accuracy, acoustic parameters must be well chosen according to the assumption of the continuous probability density functions.

B. Discrete

In the discrete HMM, the discrete probability distributions are sufficiently powerful to model any random events with a reasonable number of parameters. In this approach, the entire acoustic space is divided in to moderate number of regions, by a clustering procedure known as VQ. The centroid of each cluster is represented by a scalar codeword, which is an index in to a codebook that identifies the corresponding acoustic vector. Each input frame is converted to a codeword by finding the nearest vector in the codebook (cited by Kassehun, 2010). The HMM output symbols are also code words. Thus, the observation probability distribution over acoustic space is represented by a simple look-up table over the codebook entries.

C. semi Continuous

The semi-continuous hidden Markov model (SCHMM) has been proposed to extend the discrete HMM by replacing discrete output probability distributions with a combination of the original discrete output probability distributions and continuous probability density functions of a Gaussian codebook. In the SCHMM, each VQ codeword is regarded as a Gaussian probability density (R. Raboner, 1989). Intuitively, from the discrete HMM point of view, the SCHMM tries to smooth the discrete output probabilities with multiple codeword candidates in VQ procedure. From the continuous mixture HMM point of view, the SCHMM ties all the continuous output probability densities across each individual HMM to form a shared Gaussian codebook, i.e. a mixture of Gaussian probability densities. With the SCHMM, the codebook and HMM can be jointly re-estimated to achieve an optimal codebook/model combination in sense of maximum likelihood criterion. Such a tying can also substantially reduce the number of free parameters and computational complexity in comparison to the continuous mixture HMM, while maintain reasonable modeling power of a mixture of a large number of probability density functions.

4.2.5. HMM Assumption

According to (Rabiner, 1998) we need to consider three assumptions while using HMM which are made of mathematical and computational traceability. The first assumption states that the observation accurately represents the speech signal. Normally the observation takes the form of some type of short term (like 10ms) spectra. These will not exactly represent the underlying speech. However it can be said that the speech is nearly stationary over such short periods and the observation is a reasonably accurate representation of the speech. This assumption is

important in that prohibitively large amount of data would have to be processed, otherwise (Markowith, 1996).

The second assumption dictates that the likelihood of generating each observation is dependent only upon the state and is independent of all the other observations (cited by Zegeye, 2003). This is not typically true of speech since the spectrum tends to change only slowly compared to the frame rate to ensure that the parameterized observations are an accurate representation of the original signal. However augmenting the observations with derivative parameters can reduce the effect of the resulting high degree of correlation between subsequent observations. When these are used the correlation does not seem to have an adverse effect on the recognition and attempts to model the correlation explicitly have not succeeded yet.

The third assumption underscores that the state transition probabilities remain unchanged.

4.2.6. Language Model

Language modeling is a task assigning a probability to a given sequence of words (Mansikkaniemi, 2010). The language model contains rudimentary syntactic information. Its aim is to predict the likelihood of specific words occurring one after another in a certain language (Jurafsky, 2009). A speaker independent continuous speech recognition system is dependent on linguistic knowledge. Hence, incorporation of linguistic knowledge of the language in the form of language model, p (w) is very essential in continuous speech recognition system.

N-gram is the most popular language model (LM) used to illustrate how the LM assigns probability to a given input word strings. According to (Kim, 2004) an N- gram is a representation of an N-th order markov language model in which the probability of occurrence of a symbol is conditioned up on the prior occurrence of N-1. N-gram language models are traditionally used in large vocabulary speech recognition system to provide the recognizer with a prior likelihood P (W) of a given word sequence (Kasahun, 2010). It is usually derived from large texts that share the same language characteristics as expected input. This information complements the acoustic model P (W/O) that models the articulatory feature of the speakers. Together, these two components allow a system to compute the most likely inputs sequence as follows:

$$W' = \text{argmax}_w P(W/O)$$

$$W' = \text{argmax}_w P(O/W) P(W)$$

4.2.7. Acoustic Model

Speech is in essence just a sequence of different sounds. Our brains are tuned to classify these sounds into basic phonetic units, or phonemes. From a sequence of phonemes we can distinguish words. From a pattern recognition point of view, this is quite an astonishing feat considering that the brain is also able to comprehend speech produced in different environments and by different speakers. Devising an algorithm for a computer to do the same is not a trivial matter. Recording speech onto a computer and converting its' representation to numbers is however very easy and cheap with today's technology (Mansikkaniemi, 2010).

An acoustic model contains the data describing the acoustic nature of all the phonemes understood by the system. Acoustic models are built through a training process using large

quantities of transcribed audio. Once the signal has been transformed into a parameterized form it must be recognized, or decoded, and turned into the underlying sequence of symbols. This decoding process requires pattern or models against which unknown utterances can be compared (Odelly, 1995). The acoustic model, P(O/W), provides the probability that the speech data was observed for a given word sequence. In principle, the required probability distribution could be found by obtaining many examples of each word 'W' and collecting the statistics of the corresponding vector sequences. The three emitting states with no skip is used in this specific study because it considers right and left.

4.2. Tools and Techniques

HTK is the "Hidden Markov Model Toolkit" developed by the Cambridge University Engineering Department (CUED). This toolkit aims at building and manipulating Hidden Markov Models (HMMs). HTK is primarily used for speech recognition. HTK tools are designed to run with a traditional command-line style interface. Each tool has a number of required arguments plus optional arguments (Young et al, 2009). Options are always introduced by a single letter option name followed where appropriate by the option value. The option value is always separated from the option name by a space. In addition to command line arguments, the operation of a tool can be controlled by parameters stored in a configuration file.

The HTK tools are best introduced by going through the processing steps involved in building a sub-word based continuous speech recognizer. There are 4 main phases: data preparation, training, testing and analysis (Young et al, 2009). Almost all the information discussed below are from this book, HTK book.

4.2.1. Data Preparation Tools

In order to build a set of HMMs, a set of speech data files and their associated transcriptions are required. Before it can be used for training, it must be converted into the appropriate parametric form and any associated transcriptions must be converted to the correct format and use the required phone or word labels. Tools used in data preparation are:

HCopy: is used to copy one or more source files to an output file. Copying each file in this manner performs the required encoding.

HList : used to check the contents of any speech file and also convert input on-the-fly, it can be used to check the results of any conversions before processing large quantities of data.

HLEd: is a script-driven label editor which is designed to make the required transformations to label files. HLEd can also output files to a single *Master Label File (*MLF) which is usually more convenient for subsequent processing.

HLStats: gather and display statistics on label files and where required, HQuant can be used to build a VQ codebook in preparation for building discrete probability HMM system.

4.2.2. Training Tools

The second step of system building is to define the topology required for each HMM by writing a prototype definition. HTK allows HMMs to be built with any desired topology. HMM definitions can be stored externally as simple text files and hence it is possible to edit them with any convenient text editor. Alternatively, the standard HTK distribution includes a number of example HMM prototypes and a script to generate the most common topologies automatically.

With the exception of the transition probabilities, all of the HMM parameters given in the prototype definition are ignored. The purpose of the prototype definition is only to specify the overall characteristics and topology of the HMM

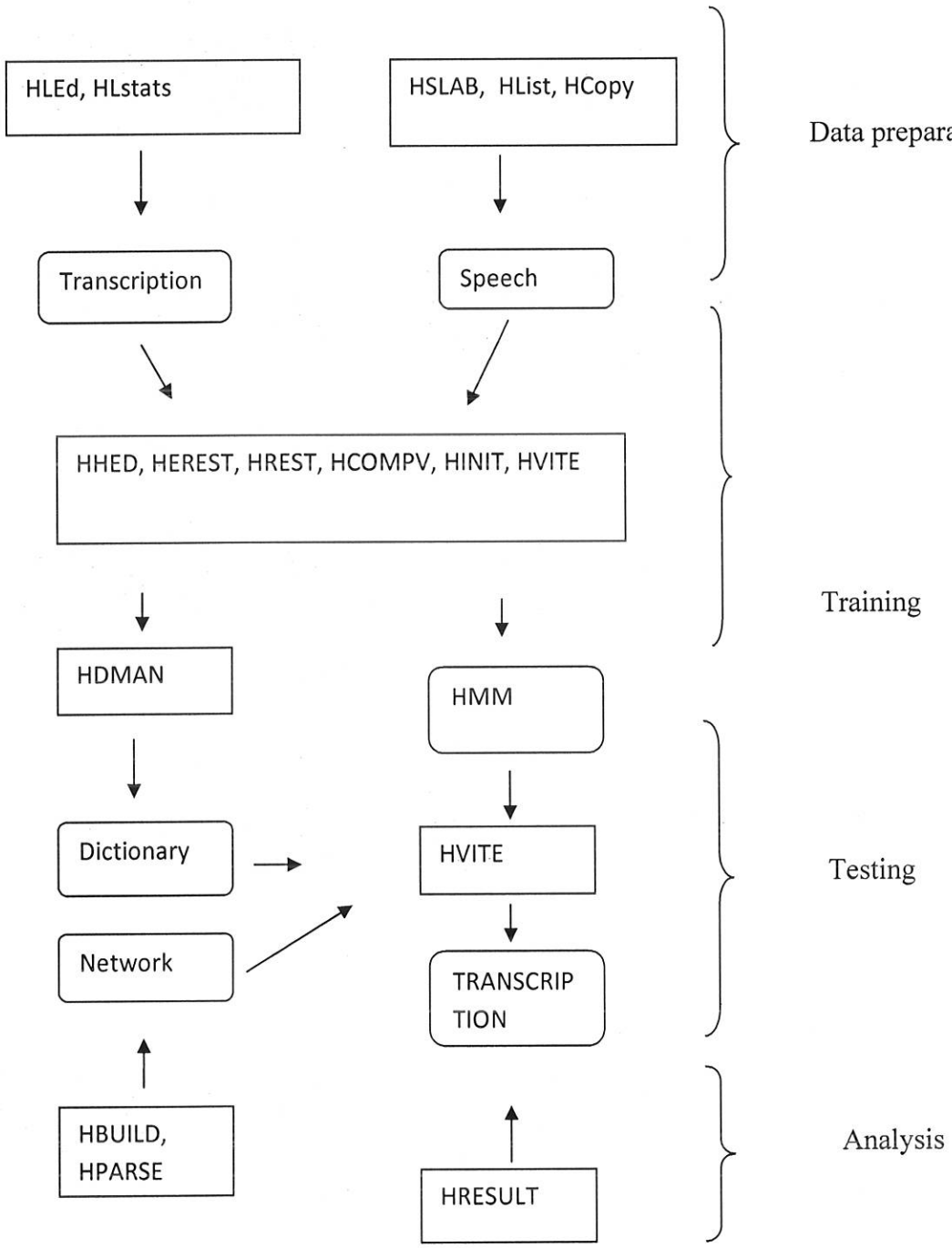

Fig.4.5. HTK processing stages

4.2.3. Recognition Tools

HTK provides a recognition tool called HVite that allows recognition using language models and lattices. HLRecsore is a tool that allows lattices generated using HVite (or HDecode) to be manipulated for example to apply a more complex language model. An additional recognizer is also available as an extension to HTK HDecode. HVite takes as input a network describing the allowable word sequences, a dictionary defining how each word is pronounced and a set of HMMs. It operates by converting the word network to a phone network and then attaching the appropriate HMM definition to each phone instance. Recognition can then be performed on either a list of stored speech files or on direct audio input.

4.2.4. Analysis Tool

Once the HMM-based recognizer has been built, it is necessary to evaluate its performance. This is usually done by using it to transcribe some pre-recorded test sentences and match the recognizer output with the correct reference transcriptions. This comparison is performed by a tool called HResults which uses dynamic programming to align the two transcriptions and then compute sentence correct, word correct and accuracy and in addition, counts substitution, deletion and insertion errors.

4.3. Multiple Pronunciations Modeling Technique

The goal of multiple pronunciation modeling is to handle multiple pronunciation that results recognition errors in ASR system (Amdal and Fosler – Lussier, 2002).

In Afaan Oromoo Certain phonemes substitute one another. This substitution creates different pronunciation from person to person which results pronunciation variation.

A corpus was collected for this study to assess this problem. The corpus has around 74 words with multiple pronunciations of the 1218 total words used.

Thus knowledge based pronunciation variation modeling technique is employed in this study. The researcher opted to use this technique because he has linguistics knowledge of the language. Thus he believes that he can handle this problem better using this modeling technique. In this approach information about the pronunciation is derived from knowledge sources such as pronunciation dictionary. However, for Afaan Oromoo there is no dictionary with multiple pronunciations. But the researcher is native speaker of Afaan Oromoo and also has linguistics knowledge of the language under study. Therefore, to develop alternate pronunciation, he used his linguistics knowledge. Hence the alternate pronunciation is developed manually by adding more entry to the canonical pronunciation dictionary when more than one possible pronunciation is available for a single word.

4.4. Testing Mechanism

Testing is an important activity to be performed while undertaking a given research. Hence, in this research the performance is evaluated against the test speech data set prepared. According to (Zegaye, 2003), accuracy of the recognizer is the most important and common parameter used to evaluate speech recognition system. Therefore, this research also uses accuracy of the recognizer in order to measure how well the recognizer classifies the input acoustic information to the corresponding text words with canonical (standard) pronunciation and alternative (multiple) pronunciations.

Following these methodologies along with appropriate tools and techniques, the next chapter shows implementation of multiple pronunciation modeling for Afaan Oromoo ASR system.

CHAPTER FIVE

Experimentation

This chapter describes the design and construction of the multiple pronunciations modeling for Afaan Oromoo ASR system. As the ultimate goal of any multiple pronunciations modeling is to assess if poor performance of ASR system is attributed to pronunciation variation, the intention of this experimentation is developing a system that can handle poor performance of automatic speech recognition due to multiple pronunciations.

The recognizer was designed to recognize continuous speaker independent continuous speech. It was implemented using phonemes as base unit. The discussion of the experiment is presented in accordance to the steps that should be followed while building multiple pronunciations modeling.

5.1. Data preparation

In order to build a set of HMMs for acoustic modeling, a set of parallel speech data files and their associated transcriptions are required. The required speech data should be prepared and made available before training the models and building the recognizer. Further, the data should pass through different preprocessing tasks like normalization, removing punctuation and giving the same name for audio and its text file in accordance to the requirements of the various algorithms (tools) in HTK.

5.1.1. Dictionary Preparation

Dictionary preparation is an important task in data preparation. For the model under construction it is obligatory to develop two types of dictionary as discussed in the methodology. These dictionaries were prepared based on triphone model in which each word is mapped to its respective phones. Using HDman tool in HTK and the produced dictionary, another two different dictionaries were created; one having short pause at the end of every pronunciation and the other without it. The global.ded script was used to attach sp at the end of each entry of the dictionary. The following table gives sample of the two dictionaries, canonical and alternate, respectively.

Word	Its transcription
Simbirroo	s i m b i rr oo
Bulchaa	b u l ch aa
Itoophiyaa	i t oo ph i y aa
Bite	b i t e

Word	Transcription
Simbirroo	s i m b i rr oo
Simbirroo	**s i n b i rr oo**
Bulchaa	b u l ch aa
Bulchaa	**b u sh aa**
Itoophiyaa	i t oo ph i y aa
Itoophiyaa	**t oo ph i y aa**

Table 5.3. Sample of canonical and alternate pronunciation dictionary

5.1.2. The phoneme Sets Extraction

Phoneme set extraction is one of the important tasks to be performed while developing the model. Accordingly, from the dictionary prepared sets of phones were extracted. The SIL phoneme was added to the phoneme list so as to handle the silence filler dictionary that causes error during training. The first, sil, models longer periods of silence. These silences occur between sentences or when a person is not speaking at all. The second phoneme, sp, also represents silence, but only periods of short duration. This is the kind of silence that occurs between words. The former represents periods of pure silence that can greatly vary in length.
It usually demarcates sentence boundaries. While the later represents optional periods of silence shorter than the duration of a phoneme, and are usually found between words.

5.1.3. File Transcription

Speech is divided into segments and each segment should have a name or a label. The set of labels associated with a speech file constitute a transcription. Two types of transcription files were expected to be prepared out of the utterances: word level and phone level transcriptions. This was because, as the system is phoneme based, later during training examples of the phones were needed to adjust the parameters of these models. That is, to train a set of HMMs every file of training data must have an associated phone level transcription.

These phone level transcriptions were created from the word level transcriptions. A program was executed to prepare the word level transcription first. It read each word from the transcription text and put them in a separate line. Then a file called Master Label File (MLF) containing all the transcriptions of the words in the utterances, indexed by the file name of the utterances was

produced. All the utterances had unique file names and the file was prepared in a label file format required by HTK. Then using the already created pronunciation dictionary and the above phone level MLF, the phone level transcription was created by the tool HLed in HTK.

5.1.4. Feature Vector Extraction

A raw audio signal received from a microphone, is too complex to deal with when it comes to the task of speech recognition. It needs to be converted into a more manageable form. This is the primary role of the feature extraction.

Feature extraction can be understood as a step to reduce the dimensionality of the input data, a reduction which inevitably leads to some information loss. Different literature reveals that Mel Frequency Cepstral Coefficients (MFCC) is one of the most commonly used feature extraction front-ends in speech recognition systems. It deals with power spectrum of speech signal which describe the frequency content of the signal over time. The purpose is to reduce the number of data characterizing the signal and shows a limited parameter or coefficient discriminating and robust.

In ASR development acoustic models are created from the already recorded speech and its transcription which is then processed to create statistical representation of the sound that make up every word. The acoustic model is later used by the speech recognition decoding engine to actually recognize the speech. Therefore, creating feature vector is essential task for creating the acoustic model.

In HTK, HCopy is a tool used for translating audio files to feature vector files using Mel scale cepstral coefficients. The feature file was created using the following script:

HCopy -A -D -T 1 -C wav config.config –S codetrain.scp.

5.2. Training HMM

After all preparation and preprocessing completed, the next step is training the model. In accordance, the model was trained in the following procedure (some of the procedures).

5.2.1. HMM Prototype

The first step in HMM training is selection of a Hidden Markov Model topology for the acoustic models. This was done by defining a prototype model. Since a phoneme based recognizer was built a model represented a phoneme. Means and variances of all the states in the model were simply assigned a value of 0 and 1 respectively. The parameters of the model are used only to define the models because they need to be modified later during training. The topology of the model consisted start and end states and three emitting states. The states were connected in a left-to-right way, with no skip transitions.

5.2.2. Initial Model

According to (Young et.al 2002), good initial models can be obtained by assuming an HMM as a generator of speech vectors. The training examples of the phones corresponding to the model whose parameters are to be estimated can be viewed as the output of this model. Thus if the state that generated each vector in the training data was known, then the unknown means and variances could be estimated by averaging all the vectors associated with each state. This is performed using HCompV tool in HTK to generate another averaged model. Hence a new

prototype was produced in a directory hmm0. This compute the global mean and variance and set all of the Gausians in a given HMM to have the same mean and variance. This was done in the following way:

HCompV –A -D –T 1 –C config.config –f 0.01 –m –S train.Scp –M hmm0 proto.proto; two files 'proto' and 'vFloor' were created by this command in the hmm0 folder.

5.2.3. Embedded Re-estimation

Once the initial monophones model set was available, the next task is re-estimating the model parameters. An embedded re-estimation strategy is used for simultaneously updating all of the HMMs in the system using all of the training data (Zegaye, 2003). The embedded procedure was implemented using the tool *HERest* in the HTK as follows.

HERest –A –D –T –C config.config –I phones0.mlf –t 250.0 150.0 1000.0 –S train.Scp –H hmm0/ macros –H hmm/ hmmdefs –M hmm1 monophones0.

5.2.4. Tied State Triphone

Neighboring phonemes influence each other while words are pronounced. To capture these effects, called coarticulation, models are needed that take into account the context of a phone. One way of modeling coarticulation effects is using triphones. Triphones model the context by taking in to consideration the left and right neighboring phones. If two phones have the same identity but different left or right context they are considered as different triphones. The HLEd command in HTK was executed to convert the monophones transcriptions in phones1.mlf to an equivalent set of triphone level transcriptions in to wintry.mlf. At the same time, a list of triphones was written to the file triphones1.

5.2.5. Realigning the Training Data

This is an important part in pronunciation variation modeling because it is at this step that words with more than one pronunciation can be recognized. As noted earlier, the dictionary contains multiple pronunciations for some words. The phone models created so far can be used to *realign* the training data and create new transcriptions. This can be done with a single invocation of the HTK recognition tool HVite, viz.

```
HVite -l '*' -o SWT -b NB_E -C config.config -a -H hmm7/macros -H hmm7/hmmdefs
 -i aligned.mlf -m -t 250.0 150.0 1000.0 -y lab -I words.mlf -S train.scp dict monophones1
>HVite_log.
```

This command uses the HMMs stored in hmm7 to transform the input word level transcription words.mlf to the new phone level transcription aligned.mlf using the pronunciations stored in the dictionary dict. The key difference between this operation and the original word to- phone mapping performed by HLEd in step 4 is that the recognizer considers all pronunciations for each word and outputs the pronunciation that best matches the acoustic data. In the above, the -b option is used to insert a silence model at the start and end of each utterance. The name silence is used on the assumption that the dictionary contains an entry silence sil.

5.2.6. Language Model

Language model predicts the most likely continuation of an utterance on the basis of statistical information about the frequency in which words, phrases and sentence sequences occur on average in the language to be recognized. It was developed using two of the HTK tools HLstats and HBuild. HLstats was primarily used to generate the bigram probability matrix. A word list and the word level MLF texts the in training were used. Using the word level transcriptions and

statistics on the number of occurrences of each word and each combination of two words, the probability matrix was prepared. These statistics were then used to create *backed-off bigram language* models for the training, test and evaluation sets, using the HBuild tool which translated the gathered statistics into HTK Standard Lattice Format.

5.3. Testing

The model trained using HTK was prepared for decoding and testing using tools HVite in HTK. Up on adjusting the parameters of the acoustic and language models of the multiple pronunciations modeling for Afaan Oromoo continuous, speaker independent speech recognizer was transferred to the appropriate directories and necessary steps were performed for testing.
The viterbi decoding algorithm implemented in HVite was used to find the utterance that maximizes the probability that a given sequence of speech sound corresponds to that utterance.

Finally the accuracy tracker performed the accuracy level obtained by comparing the text file to be recognized with the audio test sets in raw formats. To obtain this, the HResult script was executed and a performance was automatically generated with its respective statistical summary.

5.4. Analysis and Discussion of the Experiment Result

The HTK Tool HVite is a general-purpose Viterbi word recognizer. It matches speech signals against a network of HMMs and returns a transcription for each speech signal. HResults is the HTK performance analysis tool. It reads in a set of label files (typically output from the recognition tool, HVite) and compares them with the corresponding reference transcription. The perl script test.pl first calls HVite to perform speech recognition and obtain a transcription of the

test speech signals and then HResult to compute recognition statistics, percentage of correctly recognized words and the recognition statistics are displayed.

In the recognition statistics, the first line gives the sentence-level accuracy based on the total number of transcriptions generated by the recognizer which are identical to the reference transcriptions. The second line contains number of word accuracy of the transcriptions generated by the recognizer. To this end percentage of correctly recognized words is given by:

Correct $= \frac{H}{N} * 100$ where H is the number of correct words and N denotes the total number of words in the reference transcription. In addition, the tool also performed word accuracy measure which takes into account the fact that some of the words classified as correct may be in fact insertion (I) errors. This is computed by:

Accuracy $= \frac{H-I}{N} * 100$

According to the experiment undertaken for multiple pronunciations modeling of Afaan Oromoo ASR system, varying results were obtained with canonical and alternate pronunciations.

The two models have different pronunciation dictionary so as to compute their respective performance level in order to deduce whether pronunciation variation alleviates ASR performance for Afaan Oromoo or not.

The result obtained from the experiment conducted is shown below.

Sentence	%correct = 81.09 H=163 N=201
Word	%correct = 83.82 Accuracy =80.91 H= 518 D =12 S =88 I =18 N=618

Table 5.4. Recognizer Performance of Canonical Pronunciation

Sentence	%correct = 83.08 H=167 N=201
Word	%correct = 85.11 Accuracy =82.52 H= 526 D =7 S =85 I =16 N=618

Table 5.5. Recognizer Performance of Alternate Pronunciation.

Here, H is the number of correct words, D is the number of deletions (words that are present in the reference transcription, but are 'deleted' by the recognizer and do not occur in the recognizer's transcription), S is the number of substitutions (words in the reference transcription that are 'substituted' by other words in the recognizer's transcription), I is the number of insertions (words that are present in the recognizer's transcription but not in the reference), and N is the total number of words in the reference transcription.

According to table 5.4 shown above, the experiment result for canonical pronunciation indicates 81.09% sentences and 83.82 % words were correctly recognized with word accuracy of 80.91%. Using alternate pronunciation as shown in table 5.5 above, the performance of the recognizer could be increased to 83.08% and 85.11% sentences and words correct respectively with 82.52% word accuracy.

The performance of the two models was checked against the level of accuracy obtained. The model with alternate pronunciation by far weights that of model with canonical pronunciation. Furthermore, the result obtained is encouraging towards resolving problem of pronunciation variation in ASR for Afaan Oromoo.

CHAPTER SIX

Conclusion and Recommendation

Multiple pronunciations modeling for Afaan Oromoo speaker independent speech recognition was performed and varying result was obtained from the two models developed. The following section presents concluding remarks of the result obtained and forwards further future works.

6.1. Conclusion

As the intension of multiple pronunciations modeling in ASR system is to handle pronunciation variation that alleviates ASR performance, an experiment was performed in this study for Afaan Oromoo ASR system toward achieving objectives set earlier to improve Afaan Oromoo ASR system through modeling multiple pronunciations. To arrive at appropriate results for the language under study, it was performed through integration and implementation of appropriate tools, techniques and methods for the possible outcomes of the recognizer.

There are two types of multiple pronunciations modeling approach: data driven and knowledge based. The later one was implemented in this specific study.

Pronunciation variation can be modeled within word and across word; but this research considered only variation within a word and besides, it considered only Mecha dialect. Hence, a model was developed with canonical pronunciation and alternate pronunciation to check their performance against one another. The performance achieved using canonical pronunciation was 81.09% and 83.82 % correct for sentences and words respectively with word accuracy of 80.91%. Using alternate pronunciation the performance could be further improved to 83.08% and 85.11% sentences and words correct respectively with 82.52% word accuracy.

The accuracy obtained depicts by far that a model with alternate pronunciation showed better performance. Therefore, this study concludes that multiple pronunciation model can handle multiple pronunciations that alleviate Afaan Oromoo ASR performance.

6.2. Recommendation

It has been indicated that the multiple pronunciations modeling demonstrated here was only for Mecha dialect. Thus further study should be extended to model different dialects across the region, Oromiyaa.

The modeling technique employed for the multiple pronunciations in this study was knowledge based. But also it is possible to model multiple pronunciations using data-driven based; hence if further study is undertaken using data-driven approach, a better improvement may be achieved.

In addition, in this study variation was modeled only within word but we can also model pronunciation variation across word. Therefore, this is also recommended for further study.

References

Andr´e Mansikkaniemi. 2010. Acoustic Model and Language Model Adaptation for a Mobile Dictation Service. Master's thesis, Aalto University.

Annika Hämäläinen.2009. Variation in Speech: Describing Continuous Phenomena with Discrete Representation, Finland.

Ashenafi Legesse. 2009. Speaker Independent Speech Recognition for Afaan Oromoo: Unpublished Msc. Thesis, Addis Ababa University.

Anusuya.2009. "*Speech Recognition by Machine: A Review.*" International Journal of Computer Science and Information Security, Vol. 6, No. 3, Mysore, India

Bender. 1976. *The Non-Semitic Languages of Ethiopia East Lansing*: African Studies Center, Michigan State University.

*B.H. Juang and Lawrence R. Rabiner.2004.*Automatic Speech Recognition – A Brief History of the Technology Development. *Georgia Institute of Technology, Atlanta*

Crème lie. 1999. In search of better pronunciation models for speech recognition.

Cucchiarini, **C**. 1993. *Phonetic Transcription: A Methodological and Empirical Study*. Ph. D. Thesis, University of Nijmegen, Netherland.

Daniel Jurafsky. 2009. Speech and Language Processing: An Introduction to Natural Language Processing, computational Linguistics, and Speech Recognition.

Daniel Ramage. 2007. Hidden Markov Models Fundamentals: University of California, Los Angeles.

Fernando Trujillo. 2003. English Phonetics and Phonology: The Production of Speech Sounds.

Fosler-Lussier, E. 1999. *Dynamic Pronunciation Models for Automatic Speech Recognition:* Ph. D. thesis, University of Nijmegen, Netherland.

Gadaa Malbaa. 1988. An Introduction to the History of the Oromo People: Khartoum, Sudan.

Ganapathiraju. 2002. Support Vector Machines for Speech Recognition: A Dissertation Submitted to the Faculty of Mississippi State University.

Gopala krishna anumanchipalli. 2008. Modeling Pronunciation Variation for Speech Recognition: A Thesis *submitted for the award of the degree of* Master of Science. International Institute of Information Technology; Hyderabad India.

Gragg. 1982. *Oromo Dictionary:* The African Studies Center, Michigan State University.

Hemdal and Hughes. 1967.

Holmes J., Holmes W. 2001. Speech Synthesis and Recognition: Second Edition New York.

Huang. 1989. Multiple Codebooks Semi-Continuous Hidden Markov Models for Speaker-Independent Continuous Speech Recognition: Carnegie Mellon University.

http://www.sas.upenn.edu/African_Studies/Hornet/Afaan_Oromo_19777.html.

Ingunnamdal and Eric Fosler-Lussier. 2002. Pronunciation Variation Modeling in Automatic Speech Recognition. Norwegian University of Science and Technology, Norway.

John R. Deller. 2000. Discrete-time Processing of Speech Signals: Michigan State University.

Juang & Lawrence R. 1989. A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition vol.77 no 2 USA.

Judith Maria Kessens. 2002. Making a Difference on Automatic Transcription and Modeling of Dutch Pronunciation Variation for Automatic Speech Recognition; Nijmegen University, Netherland.

Kebede Hordofa. 2009. Towards the Genetic Classification of the Afaan Oromoo Dialects: PhD. Dissertation, University of Oslo.

Karen Livescu. 2005. Feature-Based Pronunciation Modeling for Automatic Speech Recognition; Massachusetts Institute of Technology.

Kasehun Gelana. 2010. Speaker Independent, Continuous Speech Recognition for Afaan

Oromoo: Unpublished Msc. Thesis Addis Ababa University.

K.N. Stevens. 1998. Production and Classification of Speech Sounds. The MIT Press, Cambridge.

Lawrence R. Rabiner.1988. A Tutor on Hidden Markov Models and Selected Applications in Speech Recognition, USA

Lee. 1987. Automatic Speech Recognition: the Development of the SPHINX system; Kluwer Academic London.

L.R. Rabiner. 1989. A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition Proc. IEEE.

Mahdi Hamid. 1995. *Oromo Dictionary: English-Oromo*, Vol.I, Atlanta, Georgia; Sagalee Oromo Publishing co.

Manish P. Kesarkar.2003. Feature Extraction for Speech Recognition, Bombay.

Mark Stamp. 2012. A Revealing Introduction to Hidden Markov Models. San Jose State University.

Mark Tatham and Katherine Mortan.2006. Speech Production and Perception, Britain.

Markowitz, J. A. 1996. Using Speech Recognition. Upper Saddle River, New Jersey: Prentice Hall, Inc.

Masaki Honda. 2003. Human Speech Production Mechanism. NTT Communication Science Laboratories, Japan.

Mirjam Wester. 2002. Pronunciation Variation Modeling for Dutch Automatic Speech Recognition; University of Nijmegen, Netherland.

Mirjam Wester and *Eric Fosler-Lussier.2002.* A Comparison of Data-derived and Knowledge-based Modeling of Pronunciation Variation; University of Nijmegen, Netherland.

Noraini Seman. 2008. Acoustic Pronunciation Variations Modeling for Standard Malay Speech Recognition. Universiti Teknologi Mara, Malaysia

Odelly. 1995. The Use of Context in Large Vocabulary Speech Recognition: Dissertation

Submitted to the University of Cambridge, London.

Octavian Cheng and Waleed Abdulla. 2005. Performance Evaluation of Front-end Processing for Speech Recognition Systems, University of Auckland.

Rabiner and Juang. 1993. Fundamentals of Speech Recognition: Englewood Cliffs, New Jersey: Prentice Hall, Inc.

Ren-Yuan Lyu, Yuang-Chin Chiang and Chun-Nan Hsu. 2005. Modeling Pronunciation Variation for Bi-Lingual Mandarin/Taiwanese Speech Recognition; the Association for Computational Linguistics and Chinese Language Processing.

Solomon Gizaw. 2008. Multiple Pronunciation Model for Amharic Speech Recognition System: Msc.Thesis Addis Ababa University.

Solomon .T, Martha .Y and Wolfgang .M. 2009. Amharic Speech Recognition: Past, Present and Future. Trondheim.

Strik and Cucchiarini. 1999. Modeling pronunciation variation for ASR: a survey of the Literature. Sophia-Antopolis, France.

Peter Roach. 2009. *English Phonetics and Phonology* glossary.

V. Mantha, R. Duncan, Y. Wu, and J. Zhao. 2001. Implementation and Analysis of Speech Recognition front-ends. Mississippi State University.

Steve Young. 1996. Large Vocabulary Continuous Speech Recognition: A Review. Cambridge University.

J.R. Deller, J.G. Proakis, and J.H.L. Hansen. 1993. Discrete Time Processing of Speech Signals: Second Ed. New York, Macmillan.

Steve Young et al. 2009. The HTK Book (for HTK Version 3.4) Cambridge University.

Solomon Tefera. 2005. Automatic Speech Recognition for Amharic: Dissertationsschrift Zur Erlangung des Grades eines Doktors der Naturwisssenscaften am Fachbereich Informatik der Universtat Hambarg.

Tilahun Gamta. 1993. Qubee Afaan Oromoo: Reasons for choosing the Latin Script for Developing an Oromo Alphabet; Ororno Commentary, Vol III, No 1.

Tilahun Gamta.1998. "Consonant Clusters in Afaan Oromoo." Journal of Afaan Oromoo Studies: Volume 5. University of Manchester, England.

Wiggers. 2001. Hidden Markov Models for Automatic Speech Recognition and their Multimodal Applictions. Delft University of Technology; the Netherlands.

Woosung Kim. 2004. Automatic Speech Recognition and Statistical Machine Translation: A Dissertation Submitted to The Johns Hopkins University in Conformity with the Requirements for the Degree of Doctor of Philosophy. Baltimore, Maryland.

Zegaye Seifu. 2003. Large Vocabulary Speaker Independent Amharic ASR System: Unpublished Msc. Thesis, Addis Ababa University.

Appendix A: Afaan Oromoo Consonants (Adugna, 2012)

		Bilabial	Labiodentals	Alveolar	palato alveolar	palatal	velar	Glottal
Stops	Voiced	b		D	j		g	
	Voiceless	P		T	ch		k	'/□/
Fricative	Voiced		v	z				
	Voiceless		f	s	sh			H
Affricative	Implosive			Dh				
	ejectives	ph		x	c		q	
Nasal	Voiced	m		n	ny			
	Voiceless							
Lateral	Voiced			l				
	Voiceless							
Rhoitic				R				
Approximant	Voiced	w				y		
	Voiceless							

Appendix B: Afaan Oromoo vowels

	Front	Central	Back
Close	i		u
Mid	e		o
Open		a	

Appendix D: Program Used for Extracting Phonemes Distribution and Canonical dictionary Building.

```
fr=open("D:\SWRS\Onesmos\corpora.txt",'r')
fwend=open("D:\SWRS\Onesmos\end.txt",'w')
fwmid=open("D:\SWRS\Onesmos\middle.txt",'w')
fwbegin=open("D:\\SWRS\\Onesmos\\fbign.txt",'w')

gh1=fr.read().split()
ty={}
ty1={}
ty2={}
qube=['a','b','c','ch','d','dh','e','f','g','h','hy','i','j','k','l','m','n','ny','o','p','ph','q','r','s','sh','t','u','v','w','x','y','z']
uniq=set(gh1)
for y in qube:
    c=0
    d=0
    a=0
    for j in set(gh1):
        if y in j:
            if j.startswith(y):
                c+=1
            if j.endswith(y):
                d+=1
            if 0<j.index(str(y))<(len(j)-1):
                a+=1
    ty[y]=c
    ty1[y]=d
    ty2[y]=a

fwbegin.write("<><><><><><><><><>: List of phonems at the begining and their
frequency:<><><><><><><><><><>\n")
for k,v in ty.iteritems():
    fwbegin.write(k)
    fwbegin.write(" : ")
    fwbegin.write(str(v))
    fwbegin.write("\n")
```

```
                vocabWord.write(i[j])
                vocabWord.write(i[j+1])
            if i[j]=='c':
               if i[j+1]=='h':
                  vocabWord.write(i[j])
                  vocabWord.write(i[j+1])
            if i[j]=='p':
               if i[j+1]=='h':
                  vocabWord.write(i[j])
                  vocabWord.write(i[j+1])
            if i[j]=='s':
               if i[j+1]=='h':
                  vocabWord.write(i[j])
                  vocabWord.write(i[j+1])
            if i[j]=='n':
               if i[j+1]=='y':
                  vocabWord.write(i[j])
                  vocabWord.write(i[j+1])
               else:
                  vocabWord.write(' ')
            else:
               vocabWord.write(' ')

      vocabWord.write('\n')

vocabWord.close()
```

Appendix D: The configuration parameter used

```
TARGETKIND = MFCC_0_D_N_Z
 TARGETRATE = 100000.0
 SAVECOMPRESSED = T
 SAVEWITHCRC = T
 WINDOWSIZE = 250000.0
 USEHAMMING = T
 PREEMCOEF = 0.97
 NUMCHANS = 26
 CEPLIFTER = 22
 NUMCEPS = 12
```

Appendix E: Tree.hed

```
QS "R_dh" { *+dh }
QS "R_e" { *+e }
QS "R_ee" { *+ee }
QS "R_f" { *+f }
QS "R_ff" { *+ff }
QS "R_g" { *+g }
QS "R_gg" { *+gg }
QS "R_h" { *+h }
```

```
QS "R_hy" { *+hy }
QS "R_i" { *+i }
QS "R_j" { *+j }
QS "R_jj" { *+jj }
QS "R_k" { *+k }
QS "R_kk" { *+kk }
QS "R_l" { *+l }
QS "R_ll" { *+ll }
QS "R_m" { *+m }
QS "L_ss" { ss-* }
QS "L_sh" { sh-* }
QS "L_t" { t-* }
QS "L_tt" { tt-* }
QS "L_u" { u-* }
QS "L_uu" { uu-* }
QS "L_v" { v-* }
QS "L_w" { w-* }
QS "L_ww" { ww-* }
QS "L_x" { x-* }
QS "L_xx" { xx-* }
QS "L_y" { y-* }
QS "L_yy" { yy-* }
QS "L_z" { z-* }

TR 2

TB 350 "ST_ff_4_" {("ff","*-ff+*","ff+*","*-ff").state[4]}
TB 350 "ST_jj_4_" {("jj","*-jj+*","jj+*","*-jj").state[4]}
TB 350 "ST_y_4_" {("y","*-y+*","y+*","*-y").state[4]}
TB 350 "ST_kk_4_" {("kk","*-kk+*","kk+*","*-kk").state[4]}
TB 350 "ST_ll_4_" {("ll","*-ll+*","ll+*","*-ll").state[4]}
TB 350 "ST_z_4_" {("z","*-z+*","z+*","*-z").state[4]}
TB 350 "ST_c_4_" {("c","*-c+*","c+*","*-c").state[4]}
TB 350 "ST_pp_4_" {("pp","*-pp+*","pp+*","*-pp").state[4]}
TB 350 "ST_ch_4_" {("ch","*-ch+*","ch+*","*-ch").state[4]}
TB 350 "ST_o_4_" {("o","*-o+*","o+*","*-o").state[4]}
TB 350 "ST_w_4_" {("w","*-w+*","w+*","*-w").state[4]}
TB 350 "ST_hy_4_" {("hy","*-hy+*","hy+*","*-hy").state[4]}
TB 350 "ST_dh_4_" {("dh","*-dh+*","dh+*","*-dh").state[4]}
TB 350 "ST_sh_4_" {("sh","*-sh+*","sh+*","*-sh").state[4]}
TB 350 "ST_ss_4_" {("ss","*-ss+*","ss+*","*-ss").state[4]}
TB 350 "ST_q_4_" {("q","*-q+*","q+*","*-q").state[4]}
TB 350 "ST_qq_4_" {("qq","*-qq+*","qq+*","*-qq").state[4]}
TB 350 "ST_ph_4_" {("ph","*-ph+*","ph+*","*-ph").state[4]}
TB 350 "ST_cc_4_" {("cc","*-cc+*","cc+*","*-cc").state[4]}
TB 350 "ST_gg_4_" {("gg","*-gg+*","gg+*","*-gg").state[4]}
TB 350 "ST_v_4_" {("v","*-v+*","v+*","*-v").state[4]}
TB 350 "ST_x_4_" {("x","*-x+*","x+*","*-x").state[4]}
TB 350 "ST_h_4_" {("h","*-h+*","h+*","*-h").state[4]}
TB 350 "ST_ww_4_" {("ww","*-ww+*","ww+*","*-ww").state[4]}
TB 350 "ST_p_4_" {("p","*-p+*","p+*","*-p").state[4]}
```

```
TB 350 "ST_xx_4_" {("xx","*-xx+*","xx+*","*-xx").state[4]}

TR 1

AU "fulllist"
CO "tiedlist"

ST "trees"
```

Appendix F: The HMM Prototype

```
~o <VecSize> 25 <MFCC_0_D_N_Z>
 ~h "proto"
 <BeginHMM>
  <NumStates> 5
  <State> 2
    <Mean> 25
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 25
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
 <State> 3
    <Mean> 25
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 25
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
  <State> 4
    <Mean> 25
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 25
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
  <TransP> 5
   0.0 1.0 0.0 0.0 0.0
   0.0 0.6 0.4 0.0 0.0
   0.0 0.0 0.6 0.4 0.0
   0.0 0.0 0.0 0.7 0.3
   0.0 0.0 0.0 0.0 0.0
 <EndHMM>
```

Declaration

This thesis is my original work, has not been presented for a degree in any University and all sources of material used for the thesis have been duly acknowledged.

_____

Onesmos Amberas

This thesis has been submitted for examination with my approval as University advisor.

_____

Wondewosen T. (PhD)

_____

Sebsibe H. (PhD)