ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF INFORMATICS
INFORMATION SCIENCE DEPARTMENT

A General Approach for Amharic SPEECH-TO-TEXT Recognition

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Information Science

BY
Michael Melese
October 2009

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF INFORMATICS
INFORMATION SCIENCE DEPARTMENT

A General Approach for Amharic SPEECH-TO-TEXT Recognition

BY

Michael Melese Woldeyohannis

Signature of the Board of Examiners for Approval

_____	_____
_____	_____
_____	_____
_____	_____

DECLARATION

This thesis is my original work and has not been submitted as a partial requirement for a MSc. degree in any university.

_____

Michael Melese

August 2009

The thesis has been submitted for examination with my approval as university advisor.

_____	_____
Million Meshesha (PhD)	**Henock Leulseged**

DEDICATION

This thesis is dedicated to all people who shaped me into the person I am today, especially my beloved Family.

ACKNOWLEDGMENT

All of my efforts would have gone for nothing if it had not been for the persistent help of the Almighty GOD. Above all, I would like to express my warm gratitude to Him in the first place who makes me I am.

Next to GOD, I would like to express my heartfelt gratitude to my advisor, Dr. Million Meshesha and Henock Leulseged for his technical support, encouragement, guidance and constructive comments that you have given through out this research work. Without your constructive comment, things would have been much more difficult and the success of this work would be doubtful.

In Addition to my Advisor, I sincerely would like to acknowledge Dr. Sebsibe H/mariam for his helpful ideas and cooperation in all possible ways by devoted his precious time to provide me technical guidance.

Finally, my earnest gratitude goes to the people who have supported me during my stay in the University and encouragement of my family, friends and relatives are worth being acknowledged.

Table of Content

CHAPTER ONE 1
Introduction 1
1.1 Background 1
1.2 Speech Recognition 2
1.2.1 Challenges and Status of Speech Recognizer 4
1.3 Statement of the problem and Justification 5
1.4 Objective of the study 7
1.4.1 General objective 7
1.4.2 Specific objectives 8
1.5 Methodology 8
1.5.1 Review of related literature 8
1.5.2 Data collection and Preparation 9
1.5.3 Development Tool 9
1.5.4 Testing Procedure 10
1.6 Scope and Limitation of the study 10
1.7 Application and Uses 11
1.8 Organization of the thesis 12
CHAPTER TWO 14
Review of Related Literature 14
2.1 Introduction 14
2.2 Amharic Language 14
2.2.1 Consonant 15
2.2.2 Vowel 17
2.3 The Human Speech Production System 18
2.4 Speech-to-Text and Text-to-Speech 20
2.4.1 Digital Signal Processing (DSP) 21
2.4.1.1 Spectrographic Analysis/Sound Spectrograph 22
2.4.2 Natural Language Processing (NLP) 24

2.4.2.1 Text Analysis 25
2.4.2.2 Phonetic Analysis 26
CHAPTER THREE 27
Speech Recognition Technique 27
3.1 Introduction 27
3.2 Techniques of Speech Recognition 27
3.3 Hidden Markov Model 30
3.3.1 Basics of HMMs 31
3.3.2 HMMs for Speech Recognition 32
3.3.3 Three problems of HMM 34
3.4 The HTK 40
3.4.1 HTK Software Architecture 41
3.4.2 The toolkit 42
3.4.2.1 Data Preparation Tools 43
3.4.2.2 Training Tools 44
3.4.2.3 Recognition or Testing Tools 47
3.4.2.4 Analysis Tools 48
Chapter 4 49
Experimentation 49
4.1 Introduction 49
4.2 Experiment 50
4.2.1 Data Preparation 52
4.2.1.1 The Pronunciation Dictionary 52
4.2.1.2 Recording the Data 54
4.2.1.3 Labeling 54
4.2.1.4 Creating the Transcription Files 56
4.2.1.5 Coding the Data 58
4.4.2 Training 59
4.4.2.1 HMM Prototype 59
4.4.2.2 Fixing the Silence Models 62
4.4.2.3 Realigning the Training Data 64

4.4.2.4 Making Triphones from Monophone 65
4.4.2.5 Triphone Construction 66
4.4.2.6 Making Tied-State Triphones 68
4.4.3 Experiment results 70
CHAPTER FIVE 77
Conclusions and Recommendations 77
5.1 Introduction 77
5.2 Conclusions 77
5.3 Recommendations 79
Reference 80
Appendix 86

List of Figures

Fig. 2.1 Amharic voice along with their feature ----------16
Fig. 2.2 The human vocal organs ----------19
Fig. 2.3 Simple text-to-speech procedure----------20
Fig. 2.4 Simple Speech-to-Text procedure ----------20
Fig. 2.5: Spectrogram representation for the utterance “Bilen” ----------23
Fig. 2.6: Spectrogram representation for the utterance “Bilen” ----------24
Fig. 3.1 Speech recognition techniques ----------29
Fig. 3.2: Message Encoding/Decoding ----------33
Fig. 3.3: HTK software Architecture ----------38
Fig. 3.4: Four Stages of HTK Processing----------39
Fig. 3.5: Training sub-word HMMs ----------42
Fig. 4.1 design of the experiment ----------47
Fig. 4.2: labeling a utterances ----------51
Fig. 4.3: Fixing Silence Models----------59

List of Tables

Table 2.1Amharic consonant along with their feature: ---------- 16
Table 4.1Recognition result statistics: ---------- 69
Table 4.2 Experimental result ----------70

ACRONYM

DSP	Digital Signal Processing
FFT	Fast Fourier Transform
HMM	Hidden Markov Model
HTK	Hidden Markov Toolkit
LPC	Linear Prediction Coefficients
MFCCs	Mel Frequency Cepstral Coefficients
MFCC_C	Mel Frequency Cepstral Coefficients Compressed
NLP	Natural Language Processing
NIST	National Institute of Standards and Technology

Abstract

In this paper, the researcher has tried to investigate the capability of exploring speech recognition technique for converting an Amharic speech to text taking native and non native speaker of the language Amharic. For this, Hidden Markov Model (HMM) is used as a model along with the tool Hidden Markov Modeling Toolkit (HTK) to implement and get the desired result out of the training.

In the development process, a total of four hundred sentences are used for the training and hundred data sets which are not included in the training set are used for testing the performance of the speech recognizer.

The primary data has been collected from four different ethnic groups that could not speak out Amharic as a mother tongue and one from the mother tongue, with an input of hundred records from each group and secondary data from the previous researcher. Then, the primary data has been labeled, preprocessed, trained and realigned as per the requirement of the HTK for the purpose of training and testing the models.

During the experiment process, a lot of challenging issue pointed out which makes the researcher to draw attention in order to confront the problems that suspend the success from attainment to the point of end.

As a final point, through all this complication, the existence of the research comes to the end provided the constraint the result obtained is promising and serve as a proof that it is possible to build general speech recognition technique that convert an Amharic SPEECH-TO-TEXT using HMM.

Once the experiment is completed and result obtained, analysis on the result is forwarded through the justification that are supported by different researcher in addition to the comparison of the result obtained.

As a final point, conclusion and recommendation are forwarded for the upcoming research area in the field.

CHAPTER ONE

Introduction

1.1 Background

These days, substantial efforts have been devoted for the development of devices with fast processing speeds and large storage capacity using the current available information and communication technologies. Further, computer software of increasing sophistication, and hardware of increasing power, has opened up possibilities for enhanced entertainment opportunities on digital platforms. This includes, access to the Internet through devices such as personal computers or gaming consoles, digital television and radio applications, digital telephony etc, [50].

Through all these complex and enhanced information communication technology, the sound and text output provides communication access to people. The visually and hearing-impaired are in the middle of these group of society that requires a special form of assisting technology to easily access the available information and share their valuable information regardless of their disabilities.

The term speech-to-text service refers to a variety of systems that convert messages transmitted through sound to a text format. The text output provides communication access to individuals who cannot hear, or otherwise process auditory information directly. In the most common situation, the process involves a speaker (such as a class instructor), a person and/or a computer and specialized software for converting speech to the desired text, and a device like a laptop computer, TV monitor, LCD projector to be used by the reader to view the text in real time [58].

1.2 Speech Recognition

In 1952, as the US government-funded research began to gain momentum, Bell Laboratories developed an automatic speech recognition system that successfully identified the digits 0 to 9 spoken to it over the telephone. This is followed by major developments at Massachusetts Institute of Technology [9]. Later in 1959, they came up with a system that successfully identified vowel sounds with 93% accuracy. Then, seven years later, a system that had a vocabulary of 50 words was successfully tested and in the early 1970's, the Speech Understanding Research (SUR) program yielded its first substantial results [6].

The HARPY system, at Carnegie Mellon University, could recognize complete sentences that consisted of a limited range of grammar structures. But the computing power it required was prodigious; it took 50 contemporary computers to process a recognition channel [9].

As speech is a natural form of communication for human beings, and computers with the ability to understand speech and speak with a human voice are expected to contribute to the development of more natural man-machine interfaces [33]. This can be achieved mainly with the help of speech recognition tools.

Since the inception, speech recognition has been playing and will play an important role in future human computer interaction. In general, the field of speech recognition is a part of the ongoing research effort in developing computers that can hear and understand spoken information [37].

Speech recognition is the process by which a computer or other type of machine identifies spoken words. Basically, by means of talking to computer, and having it correctly recognizes what you are saying by understanding Utterance, Speaker dependence, vocabularies, accuracy and training [49].

In other words automatic speech recognition (ASR) system can be described as a mechanism capable of decoding the signal produced in the vocal and nasal tracts of a

human speaker into sequence of linguistic units contained in the message that the speaker wants to communicate [2].

Modern speech recognition has a wide variety of applications. Speech recognizer is used in telephone-based conversational agents that conduct dialogues with people by converting in to text. Speech recognizers are also important in non-conversational applications that speak to people, such as in devices that write the speech for hearing-impaired individual [6].

Speech recognition allows you to provide input to an application with your voice. Just like clicking with your mouse, typing on your keyboard, or pressing a key on the phone keypad provides input to an application, speech recognition allows you to provide input to the system by talking using different tool [23].

According to Henock [17], speech recognition helps people with disabilities such as hearing and visually impaired, and also enables individuals to teach 24 hours a day and 365 days a year. In addition, it can also be used to read E-mail message through telephone, listening music through e-mail and for applied research tool in laboratory for linguistics.

According to Victor [50] expression, Speech recognition is a process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words. The recognized words can be the final results, as for applications such as commands and controls or can serve as the input to further linguistic processing in order to achieve understanding of speech.

All the above mentioned different scholars are expressing the general concept of speech recognition even if they are expressing in different ways.

Having the above mentioned facts in mind, speech to text conversion has the following advantages [21]:

The first advantage is that audio forces linearity on the user. This is not something that you can do with audio; you really have to start from the starting point and run all the way

to the ending point. This could be an advantage or it could be a disadvantage, but for short ideas like a quick note, this should be an advantage.

The *second advantage* is that dictating prevents multi-tasking; that is, when you are doing your audio recording, you cannot be working on all things at the same time. Like listening music, while writing a paper as most people do. Setting up the recording program means that you will do only dictating for a while.

The *third advantage* of audio is that it removes a barrier of entry for developing an idea. Sometimes peoples feel too lazy to fully develop an idea into a document because it's just too complicated. In dictating, you don't have worry about spelling, structure and you can just simply jump around and take your idea down as soon as possible for smaller document.

A *fourth advantage* of dictating is that it is actually hands-free; so you could do something if you really have to multi-task while you are dictating. So, in extreme case of both multitasking and do something that requires you to use your hands and dictating one idea to your computer using portable voice recorder.

Speech-to-text conversion is supporting human being with hearing inability as well as it enables individual in sharing information with the community in unrestricted at large.

1.2.1 Challenges and Status of Speech Recognizer

Today, speech recognition research is interdisciplinary, drawing upon work in fields as diverse as mathematics, computer science and electrical engineering [20]. Within these disciplines, pertinent work is being done in the areas of acoustics, computer algorithms, linear algebra, pattern recognition, phonetics, probability theory, signal processing, and syntactic theory.

Despite the current status, Speech is a complex audio signal, made up of a large number of component sound waves that can easily be captured in wave form, transmitted and reproduced by common equipment such as telephone [15]. And once we move up the complexity scale and try to make a computer understand the message encoded in speech,

the actual wave form is unreliable because different sounds can produce similar wave forms and a mathematical representation of the entire signal is too large to manipulate in real time.

The larger the vocabulary, the easier it is to confuse a recognizer if the vocabulary contain words pronounced the same way but have different meanings and the presence of an unusual accent [27] [20]. In addition, Speech recognition works best in quiet, controlled environments and performs poorly in noise, especially with crosstalk from other speakers. Since automated speech recognition degrades rapidly and it requires large processing capability as noise increases.

In order to triumph over these challenges, most modern speech recognition uses probabilistic models to understand a sequence of sounds. Hidden Markov models, in particular, are used to recognize speech. To increase accuracy in speech recognition, the language models are used to capture the information that certain combinations are more likely than others [20].

1.3 Statement of the problem and Justification

It is no dubious that human beings from the early days on have gone through various social and technological developments to achieve their goal. The development of machine/technologies to the service of man in particular, has been growing so high and fast that is difficult to believe [53].

People with disabilities such as blind and deaf are another part of the population that benefit from using speech recognition system [57]. It is especially useful for people who are hand-capped, in one way or another, from mild repetitive stress injuries to involved disabilities that require alternative input for support with accessing the computer. In fact, people who used the keyboard a lot and developed repetitive stress injuries became an urgent early market for speech recognition. Speech recognition can also be used in deaf telephony, such as voice-to-text voicemail, relay services, and captioned telephone.

There are many researches that are being conducted in different language including English, Chinese, French and even for the local language like Amharic, Oromiffa, Tigirgna and other.

Due to the fact mentioned above, Speech-To-Text systems are unequivocally vital for the widespread sets of problems that are related to the communication and information sharing among the individuals in all the Ethiopic languages, specifically for the language Amharic. This is due to the characteristics nature of the Amharic language representation and the corresponding symbols for the language.

The language Amharic has been for a long period, the principal literal language and the medium of instruction and school subject in primary, secondary and tertiary school of the country [8].

The difficulties mentioned above must be studied in order to minimize the amount of resource that require for the purpose of smooth communication among these individuals.

People who have hearing difficulties often have a difficulty to speak too. Therefore speech recognition gives such people the ability to communicate with people who do not understand the sign language [43]. So that, hearing individual can effectively communicate visual individual in a public.

A number of researches had been conducted in the area of speech recognition for Ethiopic languages; Some of these work includes Concatnative Text-To-Speech (TTS) synthesis for the Amharic Language [17], Hidden Markov Model Based Large Vocabulary, Speaker Independent, Continuous Amharic Speech recognition [53], Text-to-speech Synthesis of the Amharic Language [25], An experiment using HMM is also conducted for the recognition of Isolated Amharic Consonant-Vowel (CV) syllable [46], developing a speech synthesizer for Amharic language using Hidden Markov Model [3], sub-word based Amharic word recognition: an experiment using Hidden Markov Model [24], and Text-To-Speech System for Afaan Oromo [36].

However, most of the above mentioned researchers had investigated issues that are related to Text-to-Speech. Only Kinfe [24], Zegaye [53], Bereket [3] and Solomon [46]

attempt Amharic speech to text. Solomon [46] tries to experiment and investigate the Consonant-Vowel (CV) syllable recognition system for the Ethiopic language, particularly for the language Amharic, where as knife [24] were exploiting the sub-word based Amharic word recognition, while Zegaye [53] attempt to exploit the possibilities of continuous Amharic speech recognition for speaker independent using a Hidden Markov Model (HMM).

Both Zegaye [53] and Solomon [46] pointed out that the speech recognizer are developed for those whose first language is Amharic or at least to those who can speaks out the language Amharic fluently. And their main focus is on the recognition of those people whose first language is Amharic or at least to those who can speak the language Amharic fluently.

Provided the reality, speech recognition systems that have been developed are intended to serve as a mediator for human computer interaction in a real life situation.

Hence, the main aim of this research is to investigate the possibilities of designing a general approach for converting an Amharic speech-to-text given native or non-native spokesperson of the language Amharic.

This will help individuals with hearing inability and those that are not speaking Amharic as a first language can communicate effectively with the public at large, and in addition it will enable for the purpose of dictation system.

1.4 Objective of the study

1.4.1 General objective

The general objective of the study is to investigate the possibility of designing, developing a general approach for Amharic Speech-To-Text recognition system, such that visually impaired individuals can effectively communicate with those of hearing-impaired and sharing information with the public at large.

1.4.2 Specific objectives

The research has the following specific objectives in order to achieve the above mentioned general objective

- ✓ To conduct a literature review mainly on Natural Language Processing (NLP), Digital Signal Processing (DSP) and research that are done on the speech recognition that are done previously in Amharic and other language so as to have a clear understanding of how to develop speech recognizer for the Amharic language.
- ✓ To analyze, explore and select an appropriate algorithms for the purpose of designing an Amharic speech to text recognition system.
- ✓ To build a prototype of Amharic speech recognizer that convert speech of native and non-native speaker of the Amharic language to text.
- ✓ To evaluate/test the performance of the prototype recognizers on different data sets and report the results.
- ✓ To draw conclusion and recommendation in favor of upcoming research area.

1.5 Methodology

The following widespread methodology has been engaged for the success of conducting the research that has been proposed by the researcher.

1.5.1 Review of related literature

A literature review is a critical analysis of a segment of a published body of knowledge through summary, classification, and comparison of prior research studies, reviews of literature, and theoretical articles that are conducted by different researchers.

In this paper, a range of related literature in the area of Speech, Text-to-Speech and Speech-to-text system has been reviewed, including the recognition of Amharic speech-to-text. In addition, literatures that are related to Natural Language Processing, Digital Signal Processing, and Amharic language have been reviewed for conceptual understanding and that may forward the researcher to find the way to desired goal. Moreover, to investigate and better understand the problem area comprehensive review of literature from articles, books and resource from the interned are used.

1.5.2 Data collection and Preparation

Suitable time has been arranged to triumph over the problem that arise as from the absence of well suited, silent and sound uninterrupted laboratory to record and prepare a corpus dataset that have an effect on the experiments;

A set of text data that is used for the speech recognition have been obtained from the Bereket [3], with a total number of 500 sentences which are having 1758 unique words for both training and testing.

Typing errors are corrected on the selected text and further filtering done to have a text data for a better training as well as evaluating the performance of the speech recognizer.

People are selected to speak out the text that has been selected from those who can't speak Amharic as a first language with determination of their capability to speak out as a native and non native speaker from Oromo, Tigre, Gurage, Hadya and Amhara with equal distribution purposefully.

1.5.3 Development Tool

As a development tool, the Hidden Markov Model toolkit (HTK) is used for exploring the speech recognition technique which is recommended by many of the previous researcher. In addition, the Hidden Markov Model (HMM) is used for speech recognition development tools that support the tools HTK (Hidden Markov Model Toolkit) tool which has been developed specifically for recognizing speech.

Furthermore, Perl is used for text preprocessing at different level which is important for the Hidden Markov toolkit and the tool wavesurfer used for the purpose of analyzing and processing which is recommended by Bereket [3].

1.5.4 Testing Procedure

For the purpose of training as well as for testing, corpus of data that has been collected from selected ethnic groups is used after passing through different level. This recorded data are further processed in such a way that it can be supported by the tools that are used to train and finally test the performance of the recognizer.

These data are separated in to training data of which consisting eighty percent of the total data where as the remaining twenty percent for testing the data that are not included in the training data.

After the completion of the training, the Speech-to-Text system has been evaluated with the testing data to the maximum possible extent using test data to measure the performance level of recognizer developed in the research work.

1.6 Scope and Limitation of the study

First and most of all, this research work is predominantly anticipated to explore the general speech recognition approach for converting an Amharic speech to text. This includes selecting an appropriate model that convert Amharic Speech-to-Text by taking into account those individual that can speak Amharic as first language (native) and that can not speak Amharic as a mother tongue (non-native).

For this, the researcher selected five ethnic groups of which, one of the groups can speak out the language Amharic as a first language but not the remaining and each of the group encompass 100 sentences to speak out.

This is done following a series steps through preprocessing the recorded speech, extracting the feature and measuring the performance of the speech recognizer without involving linguistic and semantic analysis of the text.

One of the limitations of this study was the absence of a convenient Amharic reference and pronunciation dictionary for this recognition. The other limitation of this research is the environmental factors that have an effect on the results of the experiments because of the inconvenient recording area to favor the research due to the noise.

In addition to the above limitation, repeated power interruption and shortage of time are also the major problem of this research. Provided this limitation, a lot of effort has been exerted to boost-up the performance of the recognizer without trying to develop any specific application.

1.7 Application and Uses

Most of the task that involves interfacing with a computer can potentially use Automatic Speech Recognition (ASR). The following are the most common application area that are identified by Sami [43], Solomon T et, al [48] and Henock [17] in the area of speech recognition:

- **Applications for people with hard of hearing and visual impaired**: The most important and useful application field in speech recognition is the reading and communication aids for those people with hearing and visual impaired.

- **Educational applications:** speech recognition can be used in many educational situations. A computer with speech recognizer can teach 24 hours a day and 365 days a year. It can be programmed for special tasks like spelling and pronunciation teaching in different languages that can be used with interactive educational applications.

- **Telecommunications and multimedia service:** speech recognition enables speech message to be forwarded through E-mail after converting in to text and

through a normal telephone line speech may also be used to speak out short text messages (SMS) in ordinary mobile phones and converted to the text format of the desired destination.

- **Man-to-Machine communication:** speech recognition may be used in all kind of human-machine interactions. For example, in warning and alarm systems speech recognizer may be used to give more accurate information of the current situation by stating the exact place of alarm and also be used to receive some desktop messages from a computer, such as printer activity or received e-mail.

- **Fundamental of applied research:** can serve as a laboratory tools for linguistic processing if there exist some repeated experiment in order to provide identical result in research.

- **Dictation systems:** This system includes medical transcriptions, legal and business dictation as well as general word processing system.

- **Command and Control systems:** These are systems that use speech in order to perform some specific actions by using utterances like "Open file", and the system will respond by opening a file.

1.8 Organization of the thesis

This paper is organized in five different but interrelated chapters. The first chapter of the thesis discusses the basic concept of speech recognition including the definition, challenge and current status, general as well as the specific objective of the study. In addition, this chapter also presents the statement of the problem and justification of the study and methodology that are used to accomplish the research along with the scope and limitation of the study and finally presents application area of speech recognition.

The second chapter presents review of related literature in the area of Amharic language and text-to-speech and speech to text, speech production system along with the spectrographic representation of speech. This chapter also reviews the concept of Digital

signal processing, Natural language processing which describe the analysis of text and phonetic besides discussing related research work in the area of speech recognition.

The third chapter deals with the theories in speech recognition technique Hidden Markov model and the three basic problems of Hidden Markov Model through providing the suggested solution to overcome the problem that exists in the model and finally discuss the notion of HTK toolkit which will guide the researcher to explore the applicability of the general speech recognition technique for converting an Amharic SPEECH-TO-TEXT.

The fourth chapter deals with experimentation activities, which are under taken to implement method and technique that are described in the third chapter. In addition to this, the researcher describes the difficulties and success faced while trying to explore and implement the techniques using the tools that has been selected. As a final point of this chapter the researcher presents the result achieved after training the model and compares and contrast the result of the experiment in order to conclude from the finding and present recommendation for future.

The fifth chapter which is the last chapter summarizes the over all task performed in the previous chapter and finally provides recommendation for the upcoming research area in the field of speech recognition.

CHAPTER TWO

Review of Related Literature

2.1 Introduction

Language is a means of communication for sharing ideas by means of speech or verbal communication with a well defined set of applications, and this is done through reducing the overhead that are caused by an alternative way of communication methods.

This capability without a doubt is one of the facts that have allowed the development of a society. In favor of speech, different scholars provide different meaning for the same concept in one or more forms of the same approach.

As the language Amharic, is the one and unique language having its own writing system along with script representation as shown in appendix A. that is spoken among African countries [18]. And the following section discusses the concept of Amharic Language.

2.2 Amharic Language

Amharic is one of the official government languages spoken in Ethiopia, among 73 languages which are registered in the country [1]. It has 17.4 million speaker as a mother tongue and 5.1 Million speakers as a second language according to the statistical census in 1998 [13]. It also has five dialectical variations spoken named as: Addis Ababa, Gojam, Gonder, Wollo and Menz (south shoa) [47].

Amharic is a Semitic Language of the Afro-Asiatic language group that is related to Hebrew, Arabic, and Syrian. Amharic, a syllabic language, uses a script which originated from the Geez alphabet (the liturgical language of the Ethiopian Orthodox Church).

The language Amharic which is phonetic in nature, having 33 basic characters each having 7 orders of which six of them are Consonant-Vowel (CV) combinations while the seventh is the consonant itself consisting more than 230 orthographic representation of symbolic language as shown in appendix A. Unlike Arabic, Hebrew or Syrian, the language Amharic is written from left to right. Amharic alphabets, as Chinese alphabets differ from Arabic alphabets which makes unique to Ethiopia writing system as shown in appendix B [5] [44].

Like any other language, Amharic speech communication plays an important role in generating sound through the most common classification of sound either in the form of consonant or vowel. The following section presents the human speech production system after discussing the consonant vowel classification along with the method of articulation for the language Amharic.

2.2.1 Consonant

A consonant is a sound in spoken language that is characterized by a constriction or closure at one or more points along the vocal tract. The word consonant comes from Latin meaning "sounding with" or "sounding together", the idea being that consonants do not sound on their owns, but only occur with a nearby vowel even if it does not reflect a modern linguistic understanding [55].

Each consonant can be distinguished by several features [16] [55]: These features are

- The manner of articulation of consonant, such as stops, fricatives, affricatives, liquids, nasals or semi-vowels.
- The place of articulation where in the vocal tract the articulators of the consonant act, such as bilabial, alveolar, or velar.
- The phonation method of a consonant is whether or not the vocal cords are vibrating during articulation of a consonant. When the vocal cords are vibrating, the consonant is voiced; when they're not, it's voiceless.
- The air stream mechanism is how the air moves through the vocal tract during articulation.

The phonetic representation of consonant sound in Amharic language are shown table 2.1.

		ከናፍራዊ (Labials)		ከንፈርና ጥርስ/ ከንፈር ወሰናዊ (Alveolar)		ድጻዊ (palatals)		ላንቃዊ (velars)		ትናጋ (Labio-Velar)		ማነቁርታዊ (Glottals)	
እግድ Stop	Voiceless ኢነዛሪ	p	ፕ			t	ት			k	ክ	ax	ዓ
	Voiced ነዛሪ	b	ብ			tx	ጥ			g	ግ	h	ሀ
	Glottalized ፈንጂ	px	ጵ			d	ድ			q	ቅ		
ሽልክልክ Fricatives	Voiceless ኢነዛሪ			f	ፍ	s	ስ	sx	ሽ				
	Voiced ነዛሪ					z	ዝ	zz	ዥ				
	Glottalized ፈንጂ					xx	ፅ						
ፍትግ Africatives	Voiceless ኢነዛሪ							c	ች				
	Voiced ነዛሪ							j	ጅ				
	Glottalized ፈንጂ							cx	ጭ				
ጉናዊ nasal	Voiced ነዛሪ	m	ም			n	ን	nx	ኝ				
ጉናዊ Liquids						l	ል						
ላሽ Liquids						r	ር						
ከፊል አናባቢ semi-vowel		w	ው					Y	ይ				

Table 2.1Amharic consonant along with their feature: Adopted and modified from [16] [44]

Getahun [16] states that, every language has consonant phonemes as a result of different type of air movement in human vocal organs. In a language Amharic there are 27 consonant phonemes among these the consonant ፕ /p/ : ብ/b/ : ጵ /px/ are borrowed words.

In general, there are two type of right to be heard in the consonant either in the form of voiced sound or unvoiced sound even if there exist sound that are both voiced and glottalized "ፈንጂ".

2.2.2 Vowel

A vowel is a type of sound for which there is no closure of the throat or mouth at any point where vocalization occurs. Vowels can be contrasted with consonants, which are sounds for which there are one or more points where air is stopped. In nearly all languages, words must contain at least one vowel [55].

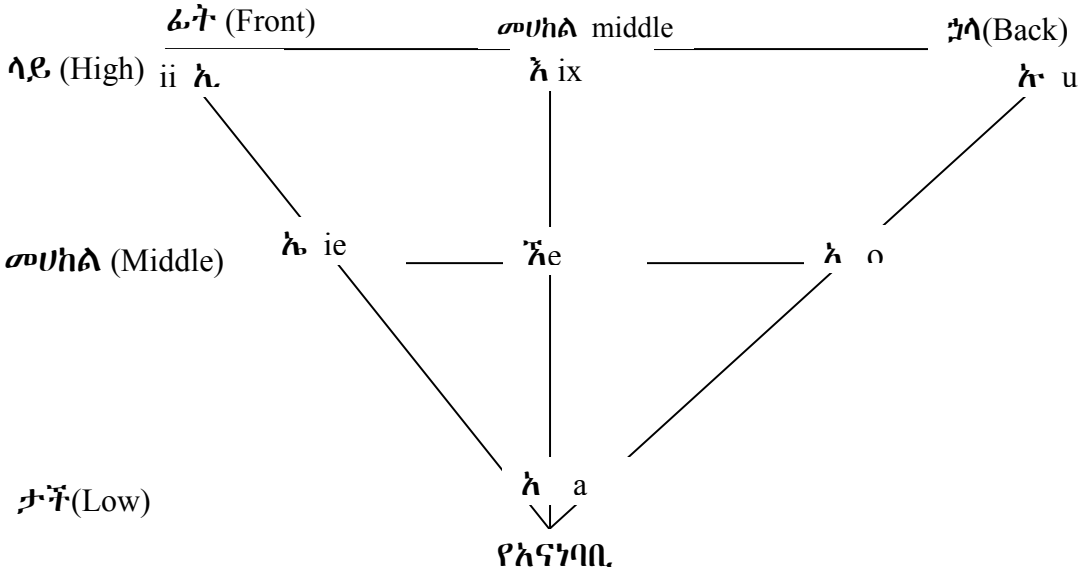

Fig. 2.1: Amharic voice along with their feature: Adopted and modified from [16] [44].

According to Solomon (46), there are 7 (seven) vowels in the language Amharic: /E/ (ኧ), /u/ (ኡ), /i/ (ኢ), /a/ (ኣ), /e/ (ኤ), /I/(እ), /o/(ኦ). Among these vowel /i/, /I/ and /u/ are high vowels. /a/ is a low vowel and the remaining three are mid vowels. Concerning the

structure of the lip while speaking, the Amharic Vowel /u/ and /o/ are rounded while the remaining /E/, /i/, /a/, /e/, and /I/ are not rounded.

The tongue is the main articulator in the mouth to determine the nature of the speech that the individual speaks-out specifically, vowel. The movement of the tongues is manipulated by a number of muscles that are available in oral cavity, which move it as a whole within the mouth. The linguistic interpretation of sound while speaking different words is done through the movement of tongue in the vertical[1] and horizontal[2] way as shown figure 2.1.

2.3 The Human Speech Production System

The principle of speech synthesis using a voice coder is to simulate the process of human speech production in the sense that the process of producing speech is expressed by the production of a sound source and reverberation of sound [33].

Human speech is produced by vocal organs presented in Figure 2.2 ranging from the diaphragm and lungs to the vocal and nasal cavities having different component in between. When speaking, the air flow is forced through the glottis between the vocal cords and the larynx to the three main cavities of the vocal tract, the pharynx and the oral and nasal cavities. From the oral and nasal cavities the air flow exits through the nose and mouth, respectively.

The V-shaped opening between the vocal cords, called the glottis, is the most important sound source in the vocal system. The vocal cords may act in several different ways during speech. The most important function is to modulate the air flow by rapidly opening and closing, causing buzzing sound from which vowels and voiced consonants are produced.

---

[1] Vertical movement of the tongue is the movement of the tongue from the high (top) level to middle or low (bottom) level or vice verse.

[2] Horizontal movement of the tongue is the movement of the tongue from the front to mid or back level or the reverse way of movement.

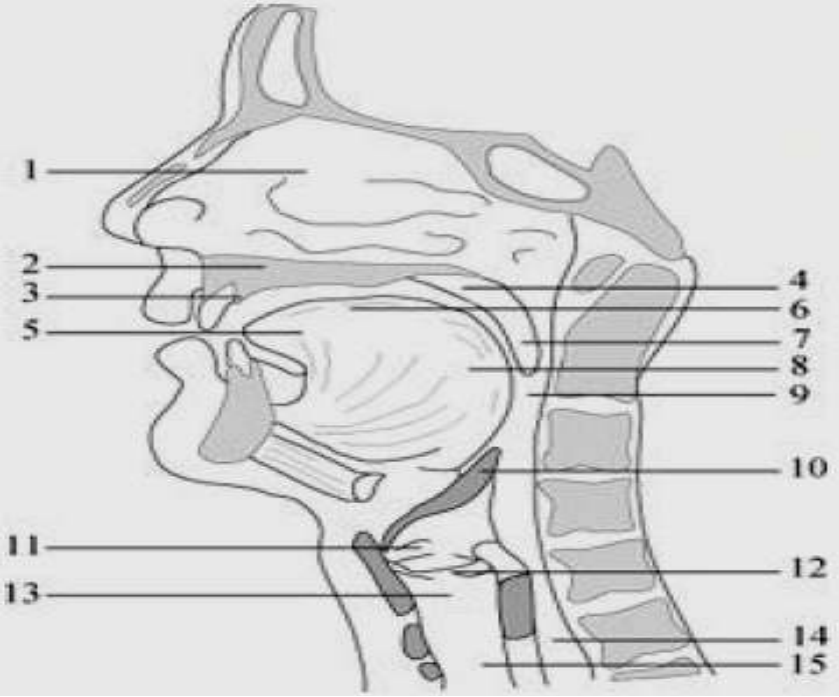

Fig 2.2 The human vocal organs. Sami L. [43]

(1) Nasal cavity, (2) Hard palate, (3) Alveolar ridge, (4) Soft palate (Velum), (5) Tip of the tongue (Apex), (6) Dorsum, (7) Uvula, (8) Radix, (9) Pharynx, (10) Epiglottis, (11) False vocal cords, (12) Vocal cords, (13) Larynx, (14) Esophagus, and (15) Trachea.

The pharynx connects the larynx to the oral cavity while the soft palate connects the route from the nasal cavity to the pharynx. At the bottom of the pharynx are the epiglottis and false vocal cords to prevent food reaching the larynx and to isolate the esophagus acoustically from the vocal tract. The epiglottis, the false vocal cords and the vocal cords are closed during swallowing and open during normal breathing.

The oral cavity is one of the most important parts of the vocal tract. Its size, shape and acoustics can be varied by the movements of the palate, tongue, lips, cheek and the teeth. Especially the tongue is very flexible, the tip and the edges can be moved independently and the entire tongue can move forward, backward, up and down. The lips control the size and shape of the mouth opening through which speech sound is radiated. Unlike the oral cavity, the nasal cavity has fixed dimensions and shape. Its length is about 12cm and volume 60cm3. The air stream to the nasal cavity is controlled by the soft palate.

2.4 Speech-to-Text and Text-to-Speech

Speech-to-text is the process of analyzing speech and producing its equivalent textual representation. Where as, Text-to-speech is the process of producing the speech for an equivalent of text given to the system [45]. Along with this, a number of research shows that Speech-to-Text (STT) is the reverse process of Text-To-Speech (TTS) and different researchers state that Speech-to-text and Text-to-speech more or less in the same way of expression.

According to Sami [43], text-to-speech (TTS) synthesis procedure consists of two phases. The first one is text analysis, where the input text is transcribed into a phonetic or some other linguistic representation, and the second one is the generation of speech waveforms, where the acoustic output is produced from the phonetic and prosodic information.

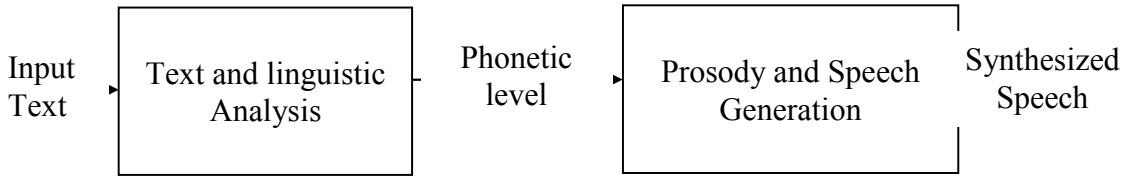

Fig. 2.3 Simple text-to-speech procedure (Sami [43])

Speech-to-text (STT) has also two phases as described in the Text-to-speech (TTS) synthesis but the tasks done in the two phases are different. In speech-to-text the first phase involves speech analysis, where the spoken input is recorded and analyzed as shown in figure 2.3, and in text-to-speech first the text is analyzed and then converted to speech through a set of procedure as shown in figure 2.4

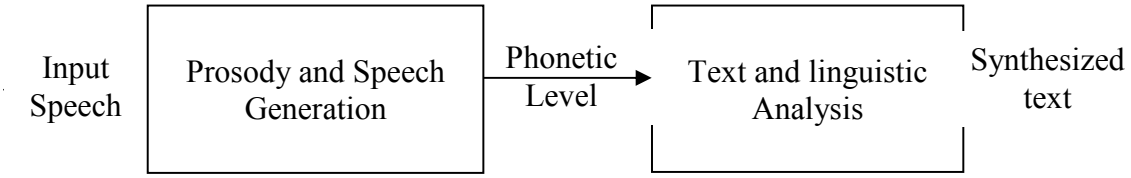

Fig. 2.4 Simple Speech-to-Text procedure (modified from Sami [43])

In both Speech-to-text (STT) as well as for Text-to-speech (TTS) system, we need to have some sort of mechanism that will enable us to arrive at the desired out put either in the form of speech or text [10][23].

Speech recognition has two modules: Natural language processing (NLP) and Digital Signal processing (DSP) [38]. The following section discusses the notion of Digital Signal Processing and Natural Language Processing.

2.4.1 Digital Signal Processing (DSP)

Digital Signal Processing is the processing of an electrical signal carried by a wire or telephones line by digital means either in the form of an analog or in digital [14]. Most signals such as sound, images and other originate as continuous-valued (or analog) signals, and must be converted into a sequence of samples to be processed using digital techniques such as Analog-to-Digital Converter (ADC). It converts instantaneous voltages of the microphone signal to corresponding numerical values, which are stored in a digital memory, and can later be sent to a Digital-to-Analog Converter (DAC) to reconstruct the original sound.

However, signal processing is not an easy task, for example, the output signal from a speaker may well be contaminated with unwanted signal/noise and processing can remove or at least reduce the unwanted part of the signal. Increasingly nowadays, the filtering of signals to get better signal quality or to extract important information is done by Digital Signal Processing (DSP) techniques [11].

Having the above motioned fact in our mind, the over all performance of any speech recognition systems is strongly influenced by the quality of the speech signal to be further processed [35].

In addition to converting the continuous signals to discrete signal values or samples that can be used to process and/or reconstruct the original signal, Digital Signal Processing (DSP) provides a number of benefits that cannot be achieved with continuous or analog signal processing [14]. These are:

- Analog hardware, such as amplifiers, filters, comparators, and the like are very susceptible to noise and deterioration through aging. Digital hardware works with only two signal levels (0's and 1's) rather than an infinite number, and hence it has a high signal to noise ratio. As a result, there is little, if any gradual deterioration of performance with age even if a digital hardware can suddenly and totally fail, and copies of signal files are generally perfect, absent component failure, media degeneration or damage, etc. which is not available in the analog hardware and recording techniques.

- Analog hardware, for the most part, must be built for each processing function needed. But in digital signal processing, each additional function needed can be implemented with a software module, using the same piece of hardware, a digital computer.

- Analog signal storage is typically redundant, since wave-related signals (audio, video, etc.) are themselves typically redundant. And by taking into account this redundancy as well as the physiological limitations of human hearing, storage needs for audio signals can be reduced up to 95%, using digitally-based compression techniques, such as MP3 and other technique.

- Digital signal processing makes possible signals to be transmitted at very low power and to share the same bandwidth through a modern cell phone techniques, such as CDMA (Code Division Multiple Access) rely heavily on advanced, digitally-based signal processing techniques to efficiently achieve both high quality and high security through error correction.

And digital signal processing further can be illustrated using the spectrographic representation and analysis in the next section.

2.4.1.1 Spectrographic Analysis/Sound Spectrograph

Spectrographic analysis is the forensic science of voice identification that has come a long way from when it was first introduced in the American courts back in the mid 1960's and comparisons were made only on the pattern analysis of a few commonly used words.

The sound spectrograph, an automatic sound wave analyzer, is a basic research instrument used in many laboratories for research studies of sound, music and speech. It has been widely used for the analysis and classification of human speech sounds and in the analysis and treatment of speech and hearing disorders by visual representation for a given set of sounds taking time, frequency and amplitude as a parameters through which the vocal sounds can be recorded and various qualities about the sounds can be analyzed [43] [34].

Articulation is one of the most relevant of all the aspects of speech production, both for the recognizer as well as for a human listener. Articulation is defined as the speed, precision, timing, and co-ordination of the separate articulators, i.e., the lips, the different parts of the tongue and the velum [30]. So that, after the articulated sound that are created by the individual will be represented using spectrographic analysis as shown in Figure 2.5 and 2.6 below for the purpose of getting a better result in speech recognition.

In the two spectrographic representations for the utterance are taken from two individual. But, the way they articulate the word "Bilen" are different from the point of the spectrographic representation shown in Figure 2.5 and 2.6.

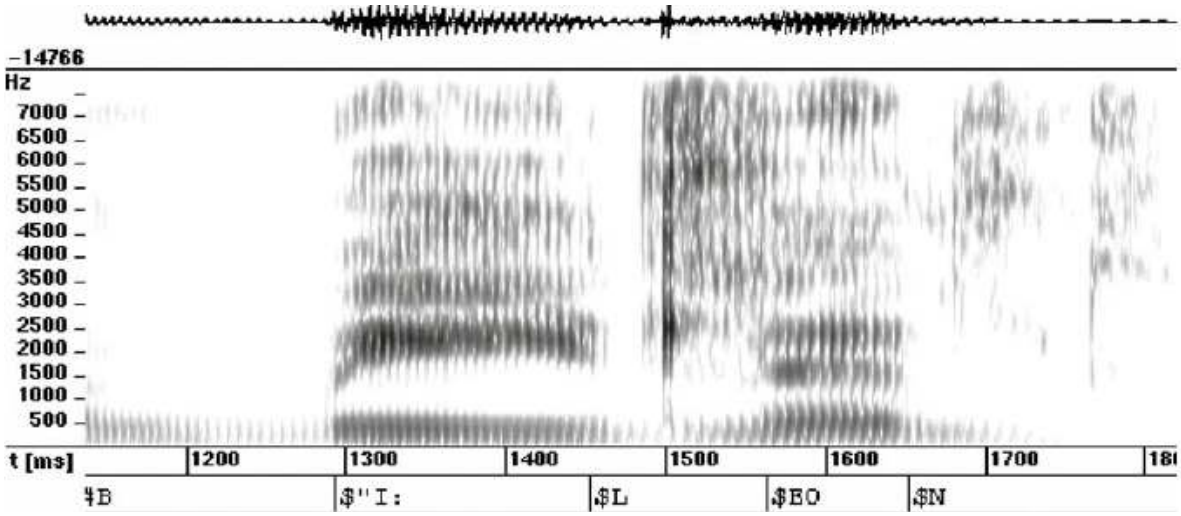

Fig. 2.5: Spectrogram representation for the utterance "Bilen": (taken from [30])

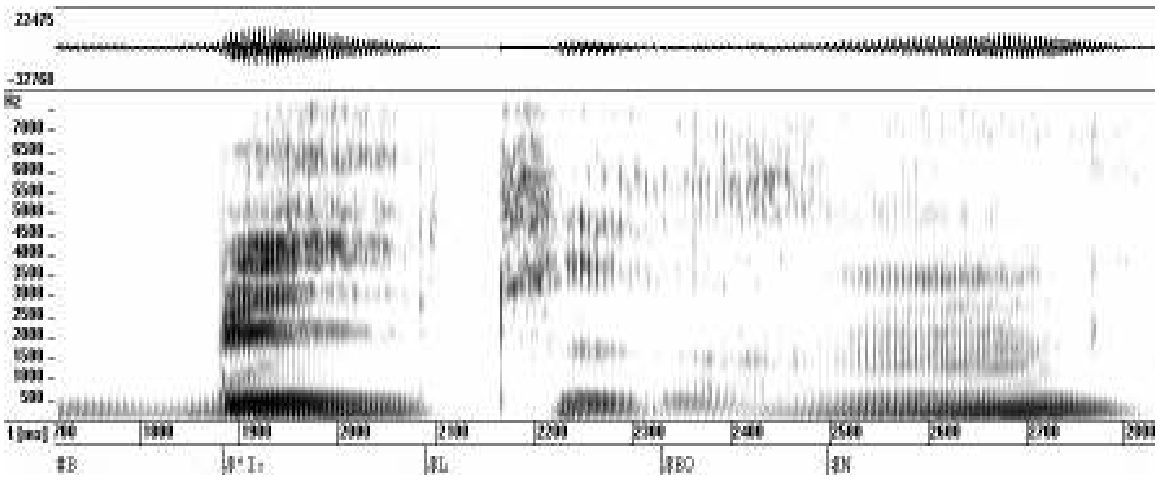

Fig. 2.6: Spectrogram representation for the utterance "Bilen": (taken from [30])

The presence of variation in spectrographic representation of utterance will result in several problems in the automatic speech recognition system; even though it is possible that some of these effects could be reduced by incorporating special adjustment to arrive in similar mode representation. Beside the spectrographic representation of the utterance, an environmental factor could also have an effect on the representation because of the inconvenient recording area due to noise.

2.4.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) does not have a single and agreed-upon definition that would satisfy everyone in the area, but there are some aspects, which would be part of any familiar goal to achieve human-like language processing.

Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks [31].

NLP can also be described as a field of computer science that concerned with the interactions between computers and human languages by converting information from

computer databases into readable human language or converting human language into more formal representations that are easier for computer programs to manipulate [54].

According to Henock [17] the main activities of natural language processing that are incorporated in speech recognition are Text and Phonetic analysis which will be discussed in the following section.

2.4.2.1 Text Analysis

Text consists of alphanumeric characters, white spaces and possibly a mixture of one or more special characters. Text analysis is one of the important stages that are presented in the process of converting speech-to-text or text-to-speech. It helps to describe the general structure and characteristics of text which will be displayed as an output or input to speech synthesis respectively.

These input and output version of text needs to be processed and converted in to linguistic representation of text in such a way that it will be suitable for further processing [38].

The first step in text analysis involves pre-processing the given input text (including expanding numerals, understanding the abbreviations and etc.) to convert it to a sequence of words [38]. During text segmentation, the character string is split into manageable chunks, usually sentences with each sentence subdivided into individual words.

Separation of words is fairly easy as words are usually delimited by white space. The detection of sentence boundaries is less straightforward. For example, a full stop can usually be interpreted as marking the end of a sentence, but is also used for other functions, such as to mark abbreviations and as a decimal point in numbers in case of English language [19]. But, when we take Amharic language (**አራት ነጥብ / ::**) is interpreted as marking the end of the statement.

2.4.2.2 Phonetic Analysis

Phonetics is one of a discipline of linguistics, which focuses on the study of the sounds that are used in speech without considering the real meaning of the sounds, the order in which they are arranged, or any other external factor of how they are formed and perceived, and their various properties [4].

Phonetics is highly related to phonology, which focuses on how sounds are understood in a particular language having three major subfields of phonetics, focusing on a particular aspect of the sounds used in speech and communication. These are auditory, acoustic and articulator phonetics.

- **Auditory phonetics:** looks at how different individuals perceive the sounds they have the sense of hearing.
- **Acoustic phonetics:** looks at the waves involved in speech sounds and how they are interpreted by the human ear.
- **Articulator phonetics:** looks at how different sounds are produced by the human vocal apparatus.

Articulator phonetics is where the greater part of people begins their study of phonetics and it has uses for many people outside of the field of linguistics which include speech therapists, computer speech synthesizers, and people who are simply interested in learning how they make the sound which will be discussed next in detail in human speech production system [4].

CHAPTER THREE

Speech Recognition Technique

3.1 Introduction

The continuous growth of information and communication technology is having an increasingly large impact on many aspects of our day to day communication through speech. And this communication is between human beings and information-processing machines which is appropriate and more important.

The main aim of this chapter is to deal with speech recognition technique which is well-known and most common modeling technique of Hidden Markov Models engaged in this research; and discusses the most important and basic concept of Hidden Markov Model. In addition this section discuss the use of Hidden Markov Model in speech recognition and finally summarizes the HMM toolkit HTK that helps the researcher to develop a prototype and implement HMM based speech recognizers with out trying to develop any specific application.

3.2 Techniques of Speech Recognition

Speech recognition is one of the most popular and researchable areas in the field of computer science. Currently many scholars are investigating the area to come up with new findings. Speech recognition systems are generally classified as discrete or continuous systems that are speaker dependent, independent, or adaptive [20].

Discrete systems maintain a separate acoustic model for each word, combination of words, or phrases and are referred to as isolated speech recognition. Continuous speech recognition systems, on the other hand, respond to a user who pronounces words,

phrases, or sentences that are in a series or specific order and are dependent on each other, as if linked together.

A speaker-dependent system requires that the user record the word, sentence, or phrase prior to its being recognized by the system. A speaker-independent system does not require any recording prior to system use. A speaker independent system is developed to operate for any speaker of a particular type. A speaker adaptive system is developed to adapt its operation to the characteristics of new speakers.

Speech is the primary means of communication between people by automatic generation of speech in the form of wave. Recent advancement in speech recognition has produced recognizer with very high simplicity but the sound quality and naturalness still stay behind a major problem in the speech recognition.

Text-to-speech is the automatic generation of a speech waveform, typically from an input text. Where as, Speech-to-text starts from a database of wave form previously collected to the typical output text [12].

In the process of text-to-speech system, text processing comes to beginning to deliver recognized speech, while in speech-to-text processing the sound processing comes to the starting so as to provide text output. In addition to this, many research shows that modern speech-to-text system requires corpus of stored data or waveforms that can be understood by the speech system beside the complexity of HMM algorithm, but naturalness remains to be improved.

In general, speech recognition passes through a number of processes to reach to the point that the desired result to be attained. For this research, the technique used to recognize speech-to-text is illustrated in the following section briefly so as to convert a speech signal into a text message

- Primarily, a set of training data is prepared that is used for the process of recognizing the speech to a set of text. The training speech file are selected and recorded as per the requirement specification as an input speech file shown in figure 3.1. Following the speech file recorded, further feature analysis is

performed on the speech so as to make it ready for the next phases provided the challenges in the absence of suitable recording area, segmentation of sound and unreliability of wave form because of different sounds can produce similar wave forms.

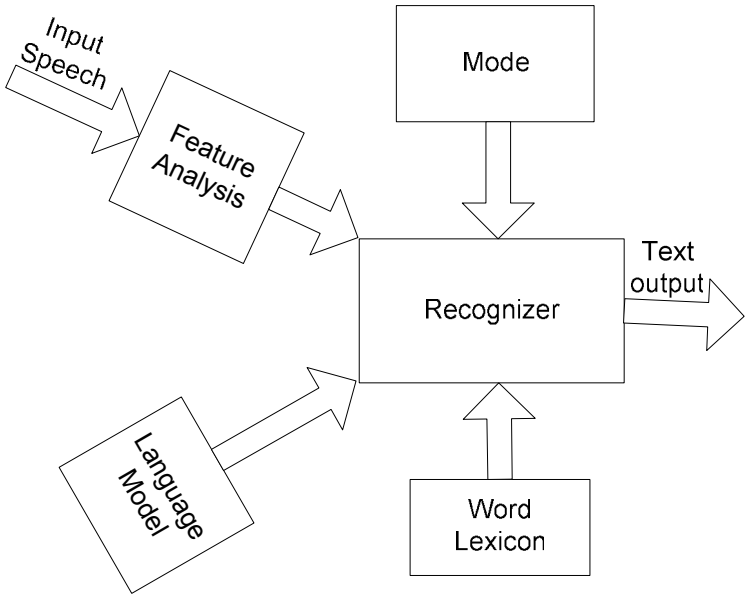

Fig. 3.1 Speech recognition techniques

Hidden Markov Model (HMM) is selected for the purpose of training the model of the speech file in such a way that can be recognized later using the testing data even though a mathematical representation of entire signal is too large to manipulate in real time. Along with this, word lexicon (tokenization and pronunciation dictionary) is also provided in addition to the selected language model for both training and testing data for the recognizer.

Once the corpus has been recorded and labeled, pronunciation dictionary constructed and appropriate language model is selected along with the acoustic model training is proceeded taking all the above as an input. Then finally the recognizer generates textual output that can be compared with the test data using the available tool as shown figure 3.1 so as achieve result.

The next section explains the Hidden Markov Model selected as model for recognizing the speech. The discussion of the Hidden Markov Model is mainly based on the instruction book that is prepared by Mark G and Steve Y [32].

3.3 Hidden Markov Model

Hidden-Markov Models (HMMs) are popular statistical models used to implement speech-recognition technologies [59]. The time variances in the spoken language are modeled as Markov process with discrete state spaces. Each state produces speech observations according to the probability distribution characteristics of that state. The speech observations can take on a discrete or a continuous value. In either case, the speech observations represent fixed time duration. The states are not directly observable, which is why the model is called Hidden Markov Model.

The foundations of modern HMM-based continuous speech recognition technology were laid down in the 1970's by groups at Carnegie-Mellon and IBM who introduced the use of discrete density HMMs. Hidden Markov Model-based technology is widely used in today's modern speech recognition systems and although the basic framework has not changed significantly in the last three decade [53].

Hidden Markov Model (HMM) is a powerful statistical tool for modeling sequences that can be distinguished by an underlying process generating an observable sequence in signal and speech processing. In addition, it can be applied with a triumph to the lower level Natural Language Processing tasks [40].

The Hidden Markov Model (HMM) statistical model provides a key benefit for the statistical approach of speech recognition. So that, the required models are trained automatically using the training data in such a way that can be recognized later using the test data prepared as per the requirement of the tool.

3.3.1 Basics of HMMs

The use of HMM's for speech recognition has be come popular in the last three decade [22]. Although the number of reported recognition systems are based on HMM's which is too large to discuss in detail here, it is worthwhile to point out some of the most important points for the success of these research. This section briefly tries to introduce the basics of HMMs, a detailed description of the model can be found in Mark G and Steve Y [32].

In most state of the art recognition systems, the Hidden Markov Model (HMM) is used in the acoustic modeling technique [53].

Hidden Markov Model (HMM) is a result of an attempt to model the speech generation statistically [39], which are the most important components of a large vocabulary continuous speech recognizer that are shown in mathematical equation on (1) shown below.

The input audio waveform from a microphone is converted into a sequence of fixed size acoustic vectors which most of recognition systems use. $Y_{1:T} = y_1, \ldots, y_T$ in a process to provide a compact representation of the speech waveform called feature extraction. The decoder then attempts to find the sequence of words $w_{1:L} = w_1, \ldots, w_L$ which is most likely to have generated Y, i.e. the decoder tries to find

$$W = \arg_w \max \{P(w|Y)\} \text{ ----------(1)}$$

However, since P(w|Y) is difficult to model directly[3], Bayes' Rule is used to transform the above equation (a) into the equivalent problem of finding:

$$w = \arg_w \max \{P(Y|w)\, P(w)\}$$

[3] There are some systems that are based on discriminative models where P(w|Y) is modeled directly, rather than using generative models, such as HMM.

The likelihood p(Y|w) is determined by an acoustic model and the prior P(w) is determined by a language model.

3.3.2 HMMs for Speech Recognition

HMMs are generative models and although HMM-based acoustic models were developed primarily for speech recognition, it is relevant to consider how well they can actually generate speech. This is not only of direct interest for recognition applications where the flexibility and compact representation of HMMs offer considerable benefits, but it can also provide further insight into their use in recognition.

A Hidden Markov Model λ (A, B, π) is defined by its parameters [51]:

- A is the state transition probability;
- B is the output probability and
- π is the initial state probability.

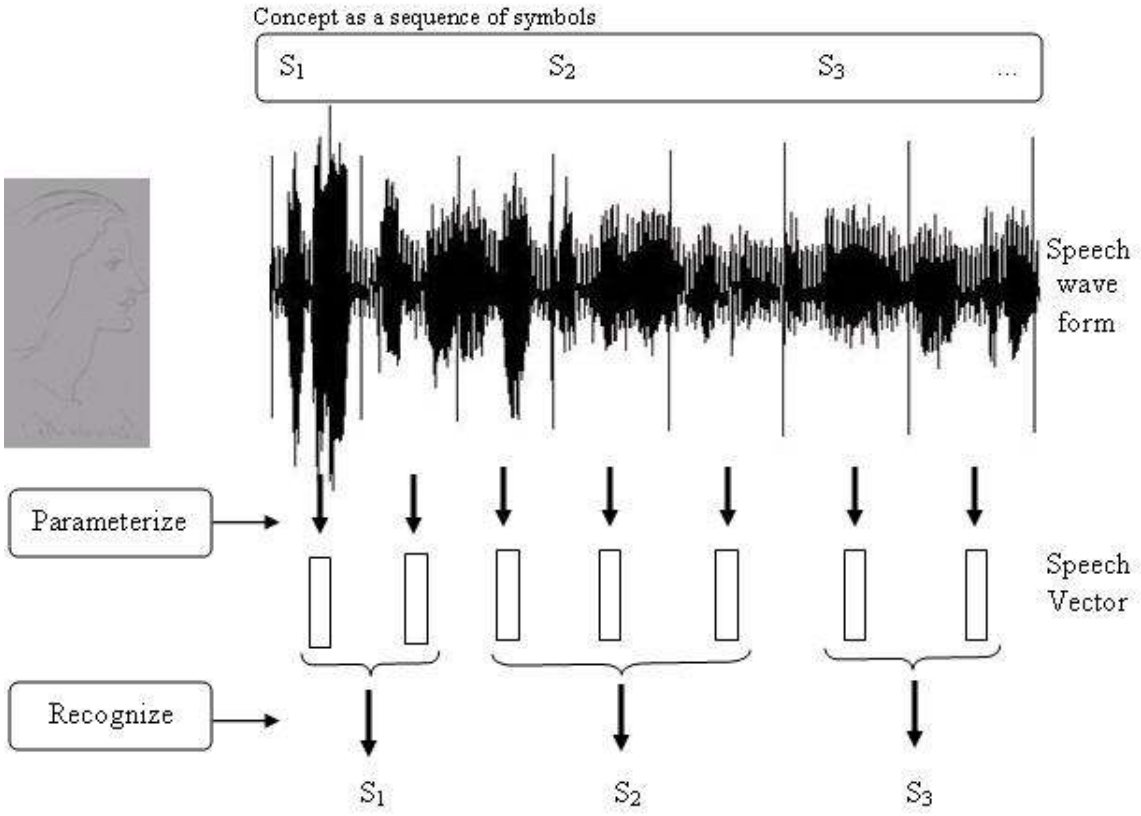

Fig. 3.2: Message Encoding/Decoding (Young [52])

The model that this section describes is a special type of HMM which is normally used in speech recognition even if it is not obvious on how the HMM related to the speech signal modeling.

Speech recognition systems generally assume that the speech signal is a realization of some message encoded as a sequence of one or more symbols as shown in figure 3.2.

The continuous speech waveform is first converted to a sequence of equally spaced discrete parameter vectors. This sequence of parameter vectors is assumed to form an exact representation of the speech waveform [52].

The role of the recognizer is to map from a sequences of speech vectors to the desired sequence of symbol even if there is a problem of mapping. Speech is not one-to-one since different underlying symbols can give rise to similar speech sounds. Furthermore, there are large variations in the realized speech waveform due to speaker variability, mood, and

environment in addition to the boundaries between symbols that cannot be identified explicitly from the speech waveform.

3.3.3 Three problems of HMM

In order for the Hidden Markov Models (HMM) to be useful in building real-world applications of speech recognizers, the following three fundamental problems of Hidden Markov Model must be solved [22]. Evaluation problem can be used for word recognition and decoding problem is related to the continuous recognition as well as to the segmentation and finally the learning problem must be solved, if we want to train an HMM for the subsequent use of recognition tasks and a detailed solution description can be found from [39].

Problem 1: Given an HMM model $\lambda = (P, B, \pi)$ and a sequence of observations $O = o_1, o_2, \ldots, o_T$, what is the probability that the observations sequence generated by the model $P(O|\lambda)$ is the problem of Evaluation**.** This problem mainly relates to evaluating how well a given model matches to a given observation and in order to solve this problem the forward algorithm is used.

In order to overcome the evaluation problem having a model $\lambda = (P, B, \pi)$, and a sequence of observations O ($o_1, o_2 \ldots o_T$), the $P(O|\lambda)$ must be found. This can be calculated using simple probabilistic arguments. But this calculation involves number of operations in the order of $\mathbf{N^T}$. This is very large even if the length of the sequence, T is moderate.

Therefore we have to look for another method for this calculation. Fortunately there exists one which has a considerably low complexity and makes use an auxiliary variable, $\alpha_t(i)$ called forward algorithm.

The forward variable is defined as the probability of the partial observation sequence $o_1, o_2 , \ldots o_T$, when T terminates at the state i. mathematically,

$$\alpha_t(i) = p\{o_1, o_2, \ldots, o_t, q_t = i|\lambda\} \quad \text{---------- (1)}$$

Then it is easy to calculate using recursive relationship and the required probability is given by,

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij}, \ 1 \leq j \leq N, \ 1 \leq t \leq T-1 \quad \text{------ (2)}$$

Where,

$$\alpha_1(j) = \pi_j b_j(o_1), \ 1 \leq j \leq N \quad \text{---------- (3)}$$

Using this recursion we can calculate

$$\alpha_T(i), \ 1 \leq i \leq N \quad \text{---------- (4)}$$

and then the required probability is given by,

$$p\{O|\lambda\} = \sum_{i=1}^{N} \alpha_T(i) \quad \text{---------- (5)}$$

The complexity of this method, known as the forward algorithm is proportional to N^2T, which is linear with respect to T whereas the direct calculation mentioned earlier, had an exponential complexity to solve evaluation problem having a model λ= (P, B, π), and a sequence of observations O (o_1, o_2. . . o_T). In a similar way we can define the *backward variable* $\beta_t(i)$ as the probability of the partial observation sequence O (o_{t+1}, o_{t+2}, o_{t+3}, . . . o_T), given that the current state is *i*. Mathematically ,

$$\beta_t(i) = p\{o_{t+1}, o_{t+2}, \ldots, o_T | q_t = i, \lambda\} \quad \text{---------- (6)}$$

As in the case of $\alpha_t(i)$ there is a recursive relationship which can be used to calculate $\beta_t(i)$ efficiently.

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(o_{t+1}), \ 1 \leq i \leq N, \ 1 \leq t \leq T-1 \quad \text{-- (7)}$$

where,

$$\beta_T(i) = 1, \ 1 \leq i \leq N \quad \text{---------- (8)}$$

Further we can see that,

$$\alpha_t(i)\beta_t(i) = p\{O, q_t = i|\lambda\},\ 1 \le i \le N,\ 1 \le t \le T \quad \text{---------- (9)}$$

Therefore this gives another way to calculate P(O|λ) , by using both forward and backward variables as given in equation 10.

$$p\{O|\lambda\} = \sum_{i=1}^{N} p\{O, q_t = i|\lambda\} = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \quad \text{---------- (10)}$$

Equation 10 is very useful, especially in deriving formulas required for gradient based training in which the detail can be found from [39].

So that the evaluation problem is solved using the forward algorithm given the model and sequence of observation.

Problem 2: The second problem of Hidden Markov Model (HMM) is decoding. Given the observation sequence O (o_1, o_2. . . o_T) and the model λ= (P, B, π), how do we choose a corresponding state sequence Q= (q_0, q_1,..., q_T) , which is optimal in some meaningful sense?

The solution to this problem depends upon the way "most likely state sequence"' is defined. One approach is to find the most likely state q_t at t=t and to concatenate all such 'q_t's. But some times this method does not give a physically meaningful state sequence.

Therefore we would go for another method which has no such problems. In this method, commonly known as Viterbi algorithm used in which the whole state sequence with the maximum likelihood is found to facilitate the computation by defining an auxiliary variable,

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} p\{q_1, q_2, \ldots, q_{t-1}, q_t = i, o_1, o_2, \ldots, o_{t-1}|\lambda\}, \quad \text{-- (11)}$$

Which gives the highest probability that partial observation sequence and state sequence up to t=t can have, when the current state is i. and it is easy to observe that the following recursive relationship holds.

$$\delta_{t+1}(j) = b_j(o_{t+1}) \left[\max_{1 \le i \le N} \delta_t(i) a_{ij} \right], \quad 1 \le i \le N, \quad 1 \le t \le T-1 \qquad \text{---------- (12)}$$

Where,

$$\delta_1(j) = \pi_j b_j(o_1), \quad 1 \le j \le N \qquad \text{---------- (13)}$$

So the procedure to find the most likely state sequence starts from calculation of $\delta_T(j)$, $1 \le j \le N$ using recursion, while always keeping a pointer to the ``winning state'' in the maximum finding operation. Finally the state j*, is found where

$$j^* = \arg \max_{1 \le j \le N} \delta_T(j), \qquad \text{---------- (14)}$$

Starting from this state, the sequence of states is back-tracked as the pointer in each state indicates. This gives the required set of states and the whole algorithm can be interpreted as a search in a graph whose nodes are formed by the states of the HMM.

Problem 3: The third problem is to determine the learning parameter and a method to adjust the model parameters $\lambda = (P, B, \pi)$ to maximize the probability of the observation sequence given the model.

Generally, the learning problem is how to adjust the HMM parameters, so that the given set of observations called the training set is represented by the model in the best way for the intended application. Thus it would be clear to optimize during the learning process can be different from application to application. In other words there may be several optimization criteria for learning, out of which a suitable one is selected depending on the application.

For this, there are two main optimization criteria found in Automatic Speech Recognition (ASR) in literature; Maximum Likelihood (ML) and Maximum Mutual Information (MMI). This problem is in fact not possible to solve using a finite observation sequence as training data, but we can choose $\lambda = (P, B, \pi)$ such that $P(X|\lambda)$ is locally maximized using an iterative procedure such as the Baum-Welch techniques under the category of Maximum Likelihood (ML) optimization.

To describe the Baum-Welch algorithm, also known as Forward-Backward algorithm, we need to define two more auxiliary variables, in addition to the forward and backward variables defined in a previous section of evaluation and decoding problems. These variables can however be expressed in terms of the forward and backward variables. First one of those variables is defined as the probability of being in state *i* at *t*=*t* and in state *j* at *t*=*t*+1. Formally,

$$\xi_t(i,j) = p\{q_t = i, q_{t+1} = j | \boldsymbol{O}, \lambda\} \quad \text{---------- (15)}$$

This is the same as,

$$\xi_t(i,j) = \frac{p\{q_t = i, q_{t+1} = j, \boldsymbol{O}|\lambda\}}{p\{\boldsymbol{O}|\lambda\}} \quad \text{---------- (16)}$$

Using forward and backward variables this can be expressed as,

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(o_{t+1})}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(o_{t+1})} \quad \text{---------- (17)}$$

The second variable is the a posteriori probability,

$$\gamma_t(i) = p\{q_t = i | \boldsymbol{O}, \lambda\} \quad \text{---------- (18)}$$

That is the probability of being in state *i* at *t*=*t*, given the observation sequence and the model. In forward and backward variables this can be expressed by,

$$\gamma_t(i) = \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}\right] \quad \text{---------- (19)}$$

One can see that the relationship between $\gamma_t(i)$ and $\xi_t(i,t)$ is given by,

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j), \quad 1 \le i \le N, \quad 1 \le t \le M$$ ---------- (20)

Now it is possible to describe the Baum-Welch learning process, where parameters of the HMM is updated in such a way to maximize the quantity, P(O|λ). Assuming a starting model λ= (A, B, π), we calculate the ' α's and 'β's using the recursions in equation 7 and 2, and then ' ξ's and 'γ 's using equation 17 and 20. The next step is to update the Hidden Markov Model parameters according to equations 21 to 23, known as *re-estimation formulas*.

$$\bar{\pi}_i = \gamma_1(i), \quad 1 \le i \le N$$ ---------- (21)

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad 1 \le i \le N, \quad 1 \le j \le N$$ ---------- (22)

$$\bar{b}_j(k) = \frac{\sum_{t=1,\, O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, \quad 1 \le j \le N, \quad 1 \le k \le M$$ ---------- (23)

These re-estimation formulas can easily be modified to deal with the continuous density case too. Then using as an initial parameter instantiation, the forward algorithm iteratively re-estimates the parameters and improves the probability that given observation is generated by the new parameters. So that, the three major problems that exist in the Hidden Markov Model are solved in such a way and the next section briefly discusses the tools used for speech recognition based on the model Hidden Markov called Hidden Markov Toolkit.

3.4 The HTK

This section tries to briefly introduce the basics of Hidden Markov Model toolkit (HTK) and a detailed discussion of the toolkit is mainly based on the instructional book that is prepared by Young S. and et al [52] and quotation will be made if otherwise.

HTK is the "Hidden Markov Model Toolkit" developed by the Cambridge University Engineering Department (CUED). The Hidden Markov Model Toolkit (HTK) is a portable toolkit for building and manipulating hidden Markov models [56]. HTK is primarily used for speech recognition research although it has been used for numerous other applications including research into speech synthesis, character recognition, gesture recognition and DNA sequencing.

HTK is in use at hundreds of sites consisting of a set of library modules and tools available in C source forms that are designed to run with a traditional command-line style interface. The layout of the commands is the same for all tools. Each tool has a number of required arguments plus optional arguments.

In addition, the operation of a tool can be controlled by set of parameters stored in a configuration file. The main use of this configuration file is to control the detailed behavior of the library modules on which all HTK tools depend.

The HTK tools can best be described in line with the steps involved to develop sub word based continuous speech recognizers. The tools provide sophisticated facilities for speech analysis, data preparation tools, training and testing along with analysis of the result given the training and testing dataset.

In HTK, there are two major processing stages involved. Firstly, the HTK training tools are used to estimate the parameters of a set of HMMs using training utterances and their associated transcriptions. Secondly, unknown utterances are transcribed using the HTK recognition tools.

3.4.1 HTK Software Architecture

Most of the functionality of HTK is built into the library modules ensuring that each and every tool interfaces to the outside world in exactly the same way. They also provide a central resource of most commonly used functions and the figure 3.3 illustrates the software structure of a typical HTK tool and showing input/output interfaces.

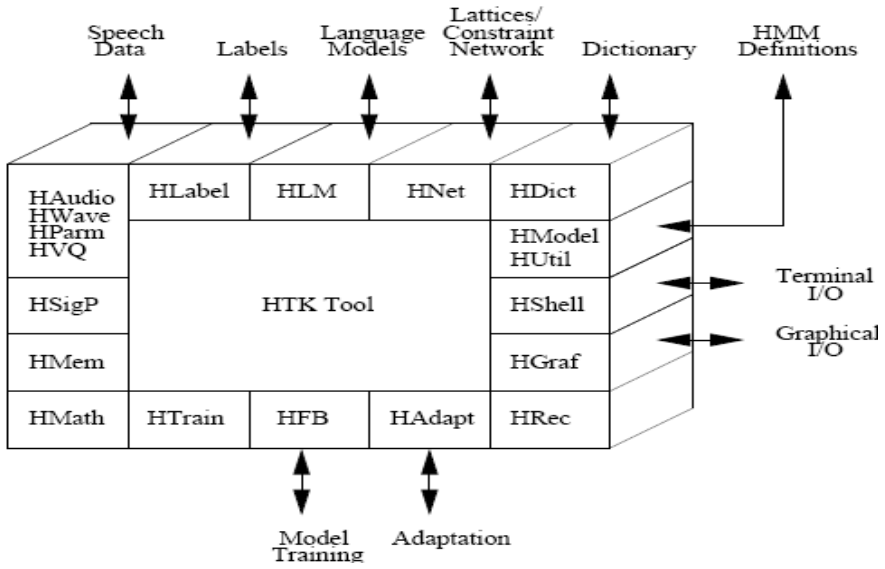

Fig.3.3 HTK software Architecture [52]

Each of the file types required by HTK has a dedicated interface module. These interfaces are:

- User input/output and interaction with the operating system is controlled by the library module HShell.
- All memory management is controlled by HMem.
- Mathematical support is provided by Hmath.
- The signal processing operations needed for speech analysis are in HSigP.

- HLabel provides the interface for label files, HLM for language model files, HNet for networks and lattices, HDict for dictionaries and HModel for HMM definitions.

In addition to the above mentioned interface module, HTK provides a lot of interface that are necessary for the success of speech recognition.

As mentioned in previous section, HTK tools are designed to run with a traditional command-line style interface having a number of required arguments plus optional arguments in which the latter are prefixed by a minus signal all the time. In addition to command-line arguments, the operation of a tool can be controlled by parameters stored in a configuration file in order to load the parameters stored in the configuration file during its initialization procedures.

3.4.2 The toolkit

The HTK tools are best described by introducing the four processing steps or phases as shown in Fig. 3.4 in building speech recognizer these four main phases are: data preparation, training, testing and analysis.

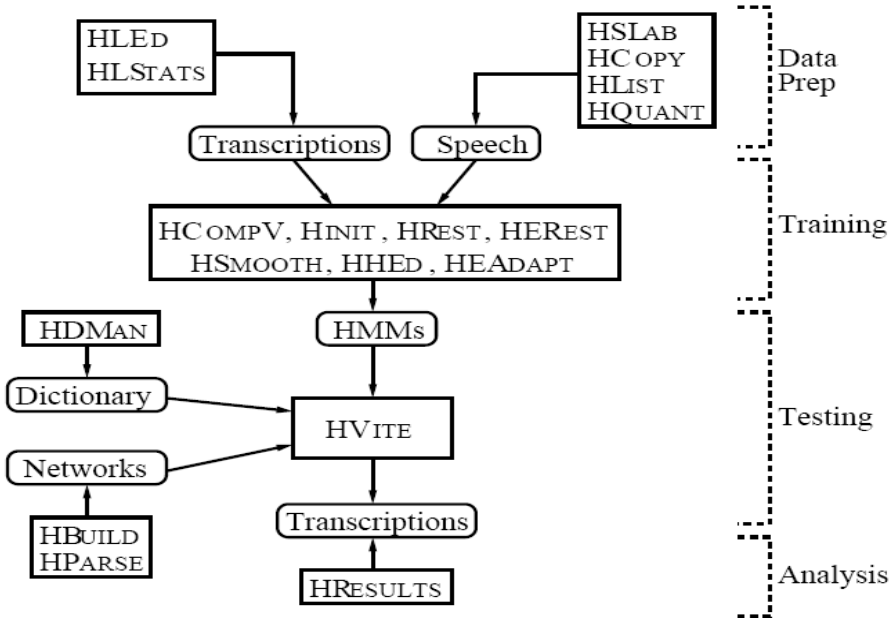

Fig. 3.4 Four Stages of HTK Processing (taken from [52])

3.4.2.1 Data Preparation Tools

Data preparation is the first stages in the process of speech recognition among the four stages of HTK Processing.

For the data preparation, different commands that are available in the HTK tools are used. Among these, HSLab, HCopy, HList and HLed are some of the interface that enables to prepare the data for the training phase of the speech recognition. And at the end of the data preparation, transcribed and the parameterized data will be readily available for the training phase.

HSLab is an interactive label editor for manipulating speech label files. Before constructing Hidden Markov Models, a set of speech data files must be prepared along with the associated transcriptions.

Most of the times in order to build Hidden Markov Model, the speech data are obtained from database archives, typically from CD-ROMs. Before using the speech corpus for the purpose of training, it must be converted into the appropriate parametric form and any associated transcriptions must be converted to have the correct format. If the speech needs to be recorded, then the tool HSLab can be used both to record the speech and to manually annotate it with any required transcriptions.

Although all HTK tools can parameterize waveforms on-the-fly, in practice it is usually better to parameterize the data just once. The tool HCopy is used for this. As the name suggests, HCopy is used to copy one or more source files to the desired an output file. Normally, HCopy copies the whole file by taking the parameter of source and destination file, but a variety of mechanisms are provided for extracting segments of files and concatenating files. By setting the appropriate configuration variables, all input files can be converted to parametric form as they are read-in. Thus, simply copying each file in this manner performs the required encoding.

HList can be used to check the contents of any speech file in addition to converting the input on-the-fly, it can be used to check the results of any conversions before processing large quantities of data. Transcriptions will also need preparing.

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.

Finally on data preparation, **HLStats** can 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.

3.4.2.2 Training Tools

Training stages is the stage in which the actual training takes place. It is the process of speech recognition to build with any desired topology by receiving the data that has been prepared by the previous stage of HTK Processing. 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 under which the actual parameters will be computed later by the training tools. The definitions file of HMM can be stored externally as simple text files and it is possible to edit them with any convenient text editor.

The following section shows how training is carried out in order to prepare an input that can be used by the speech testing tool of HTK.

An initial set of models must be created that can be modified at the later stage of the training.

- If there is speech data available for which the location of the word boundaries have been marked that can be used as bootstrap data.

- The tools HInit and HRest provide isolated word style training using the fully labeled bootstrap data and each of the required HMMs is generated individually by:

    - **HInit** reading all of the bootstrap training data and cuts out all of the examples of the required phone. It then iteratively computes an initial set of parameter values using a segmental k-means procedure by segmenting the training data in uniform size on the first cycle. Then, each model state is matched with the corresponding data segments and then means and variances are estimated.

    - On the second and successive cycles, the uniform segmentation is replaced by Viterbi alignment. The initial parameter values computed by HInit are then further re-estimated by **HRest**. Again, the fully labeled bootstrap data is used but this time the segmental k-means procedure is replaced by the **Baum-Welch** re-estimation procedure.

HcompV tool is used when no bootstrap data is available and a so-called flat start monophone can be used. In this case all of the phone models are initialized to be identical and have state means and variances equal to the global speech mean of (0) and variance of (1). The focus here is to create a model structure, the parameters are not important because, the global mean and variance are modified at the successive phase of the training.

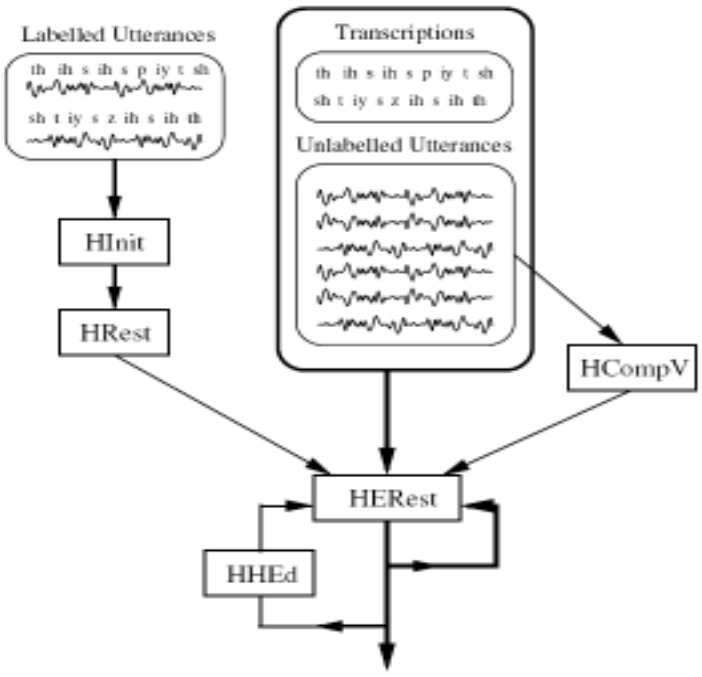

Fig. 3.5: Training sub-word HMMs [52]

HERest is one of the core HTK training tools that are designed to process large databases. **HERest** performs a single Baum-Welch re-estimation on the entire training set of the whole HMM monophone models simultaneously after the initial training set of the model created successfully as shown Fig. 3.5.

For each training utterance, the corresponding phone models are concatenated and forward-backward algorithm is used to accumulate the statistics of state occupation, means, variances, etc., for each HMM in the sequence.

When all of the training data has been processed, the accumulated statistics are used to compute and re-estimates the HMM parameters for the next phases.

In order to improve the performance of specific speakers the tools HEAdapt and HVite can be used to adapt HMMs to better model the characteristics of particular speakers using a small amount of training data.

3.4.2.3 Recognition or Testing Tools

HTK provides a number of tools that can be used for the purpose of data preparation, training, testing as well as for analysis. Each tools providing its own specific function in the areas that are mentioned in the introduction part of HTK.

The most important tool that is used in the testing section of this research is HVite tool which will be described below along with the functionalities. HTK provides a recognition tool called HVite which uses the token passing algorithm. 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.

HVite can support cross-word tri-phones and it can run with multiple tokens to generate lattices containing multiple hypotheses. It can also be configured to rescore lattices and perform forced alignments. Along with HVite there are other tools that support testing and recognition of speech and a detail can be found from the HTK book [52]: these tools

- HSGEN
- HLStats
- HDMan and
- Hparse

3.4.2.4 Analysis Tools

Once the HMM-based recognizer has been built, it is necessary and important to evaluate the over all 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 count substitution, deletion and insertion errors.

Options are also provided to ensure that the algorithms and output formats used by HResults are compatible with those used by the US National Institute of Standards and Technology (NIST). As well as global performance measures, HResults can also provide speaker-by-speaker breakdowns, confusion matrices and time-aligned transcriptions.

Chapter 4

Experimentation

4.1 Introduction

The main concern of this chapter is to describe the designing, constructing and developing of the general speech recognition technique for converting the Amharic language that is capable of recognizing Amharic spoken language.

The intention was developing a prototype that is as general as possible, so that it can be used, with slight modifications and some adaptive training as a speech interface for many other application areas.

Any systems have a number of components that interact in order to achieve some objective in one or more fashion. These interactions among the components results in generating an output that might be used as an input to another systems or a system which may also trigger another system.

Having this in mind, the development process of the speech recognition was performed on the UNIX platform using the tools in Hidden Markov Model toolkit (HTK). Along with this tool, various preprocessing tool such as wavesurfer for sound analysis and sound transcription, and scripts that exist on the UNIX environment were also used. In addition to this tool, scripts that exist in the Perl are also used for preprocessing the raw text data that has been selected.

The recognizer that is going to be developed in this paper is intended to recognize an Amharic speech. But, for the future systems this can be extended by a large set of corpus, vocabulary that consists of a set of pronunciation of the speech to meet the requirement of the new recognizer. The following section provides the experimental processes that have been carried out in this work.

4.2 Experiment

The discussion of the experiment is presented in accordance to the experiment design shown figure 4.1 while building such speech recognizer. Accordingly, the experiment design is organized in to four parts preprocessing (data preparation), Training, Recognition and finally Evaluation of the recognizer each of the phases delivering some out for the succeeding phase of speech recognition.

The first phase mainly concerned with feature extraction on the text taking the raw sentences. Since the application requires arbitrary set of sentence for both training and testing the speech recognizer.

In this experiment, 500 different sentences have been selected which is consisting of a total 1758 unique words. The text that is used for both the training sets as well as testing sets are taken from Bereket [3] which has been selected using purposive sampling from a corpus that has 11,670 sentences. This data is further processed so as to fill the requirement of tools either for training or testing the data.

Once the data transcribed in to word and syllable label, it divides in to either for training or testing the data up on the requirement of sample size. The data that are used for the training passes through different level including realignment, converting from monophones to triphones and additional rearrangement on the training so as improve the accuracy of the recognizer.

After this point the training and the testing data are given to the recognizer to measure the over all performance of the recognizer and finally the result of the recognizers are evaluated using the tool HResult.

Figure 4.1 shows the design of the experiment used for this research and each of the phases are illustrated following the diagram.

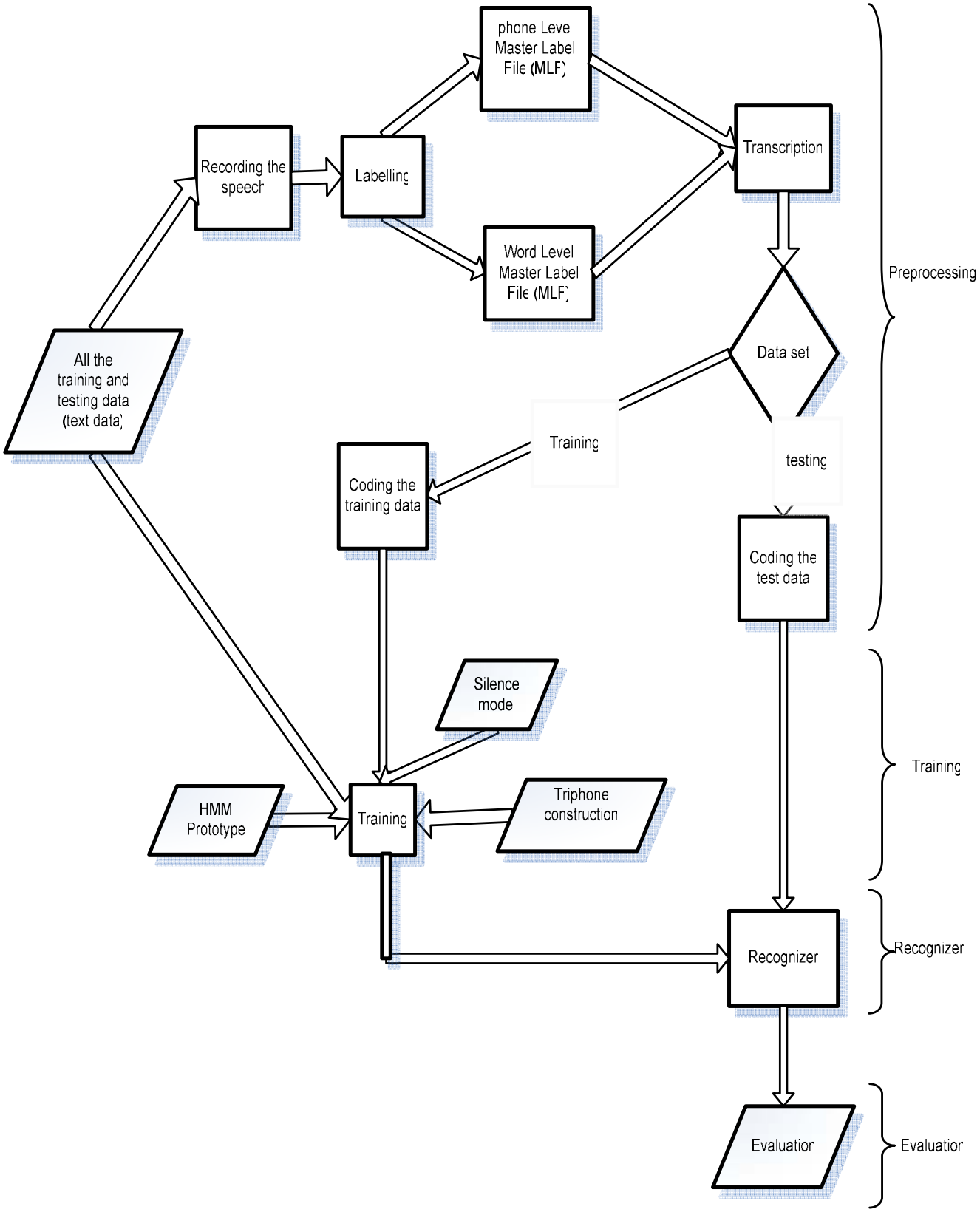

Fig. 4.1 design of the experiment

4.2.1 Data Preparation

Data preparation is the first stage of any development of speech recognizer and this section presents the process that are involved in the data preparation task through process of tokenization, pronunciation dictionary, recording the data, labeling, creating the transcription files and finally coding the data in such a way that the data is ready for the training phase of speech recognizer.

4.2.1.1 The Pronunciation Dictionary

Pronunciation refers to the way a word or a language is spoken, or the manner in which someone utters a word. A pronunciation dictionary is a file that contains all words along with the way they are pronounced.

In order to create pronunciation dictionary first you need to have a list of unique words extracted from the sample 500 sentences by tokenizing in to list of words. Tokenization is the process of demarcating and possibly classifying sections of a sentence into a list of words that are separated by a new line.

The script “promps2wlist” is used to tokenize the given sentence in to list of words that are separated by space in a given sentence. This list of word contains repeated words that are not sorted by excluding the first words from each line. Then the UNIX tool unique sort is implemented in order to make the list sorted in addition to removing duplicate data.

The expected major deliverables of this task is a raw and unique word which is important for the purpose of constructing the pronunciation dictionary that is used throughout the experiment in this paper.

Pronunciation dictionary is an important and valuable resource in the process of building speech recognizer. For the English language and other technologically supported languages, commercial and public domain pronunciation dictionaries are available. But, the language like Amharic and other languages that are not technologically supported, the pronunciation dictionary need to be constructed manually.

In order to conduct a research in a language like Amharic, which is not supported technologically, it is a must to prepare a list of pronunciation dictionary. This is done by taking list of unique words that are prepared in the earlier stage of data preparation and converting in to the form of unit of spoken language consisting of a single uninterrupted sound formed by a vowel, diphthong, or syllabic consonant alone, or by any of these sounds preceded, followed, or surrounded by one or more consonants.

The pronunciation dictionary prepared in this stage is to be used by the tool HTK has a very simply format. Each line consists of a single unique word and the corresponding pronunciation.

The general formats used for each dictionary entry is:

Sample-word [output] pr_1 pr_2 pr_3 . . .

Where pr_1, pr_2, pr_3 … are the pronunciation of the sample-word in a given utterance. The output in square brackets specifies the recognized string from a given sample word. If it is omitted then the word itself is output.

For example the Amharic word ”**አቅጣጫቸውን**” can be pronounced as

አቅጣጫቸውን **አ ቅ ጣ ጫ ቸ ው ን**

aqtxacxacewn a q txa cxa ce w n

And sample of the pronunciation dictionary shown below

siiqerexx		sii qe re xx
siirotxu		sii ro txu
siitay		sii ta y
siiyagenxu		sii ya ge nxu
siiyamesegixnuh		sii ya me se gix nu h
siiyamsen		sii ya m se n
siiyasxenfu		sii ya sxe n fu
siiyasxenixf		sii ya sxe nix f
siiyayew		sii ya ye w
siiyayunx		sii ya yu nx
sil	[]	sil

4.2.1.2 Recording the Data

Before recording the data all the text were classified randomly in a group of twenty five sentences and written in the Amharic by transliterating the sample taken and prepared in readable format.

Once all the data has been prepared and the pronunciation dictionary is constructed, for both the training and test data need to be converted in the form of sound using the tool HSLab in such a way that can be analyzed using the HTK tools. However, all the data that has been recorded by the preceding researcher Bereket K. [3] were not suitable for this work. For this, the researcher needs to record all the sound from the scratch to make suitable for this research using the available and recommended combined waveform recording tool.

HSLab is invoked by typing:

HSLab sample_num.wav

Where, sample_num.wav is the sound files to be recorded.

The database is prepared from utterances made by people whose first language is Amharic, Oromiffa, Tigray, Hadya and Gurage has been selected due to the ease of accessibility with a total sample of 500 utterances each with an equal percentage.

4.2.1.3 Labeling

In addition to recording, the tool HSLab can be used for the purpose of segmenting the sound uttered in to text based format to assign a label for each phoneme called labeling.

Once the sound recorded, labeling can be performed either automatically or manually. But generating the label without human intervention leads to an enormous error. Therefore, each and every labeling is carried out manually which is difficult, time consuming and critical for the success measure of the performance of speech recognizer.

This labeling is done using the same tool that is used for recording the speech and sample label for the utterance “beadiis abeba ketema yetekesetew yewixha ixtret”/ “**በአዲስ አበባ ከተማ የተከሰተው የውሀ እጥረት**” are shown below in figure 4.2 and the label file for this specific utterance are generated automatically as you labeling the speech recorded manually and saved with the same name as to the sound file with .lab extension.

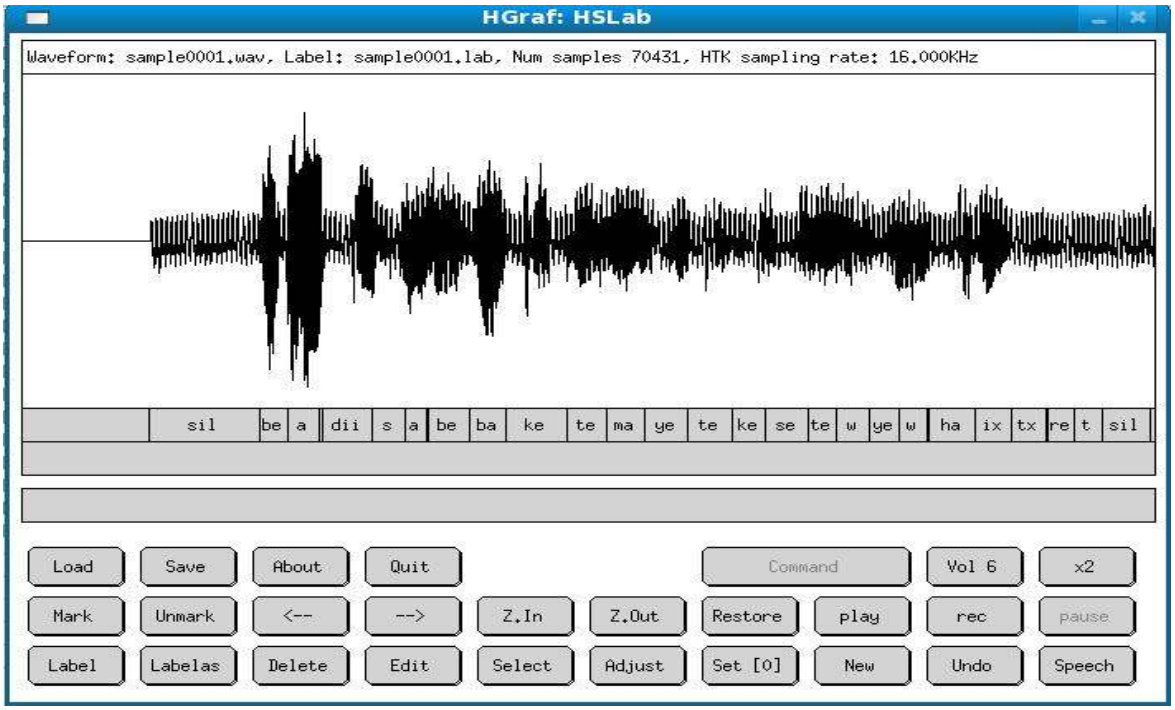

Fig. 4.2: Labeling a utterances

So that, the file .lab extension will create the following label file which is an important label information for the computing the mean and the variance in the training phase of the experiment by taking the first value, the second value and the specific utterance for the starting point, ending point and the pronunciation respectively .

5011875 9317500 sil

9317500 10410000 be

10410000 11631250 a

11759375 13687500 dii

13687500 14972500 s

14972500 15808125 a

15872500 17478750 be

17478750 18892500 ba

18892500 21270625 ke

21270625 22812500 te

22812500 24290625 ma

24290625 25961250 ye

25961250 27696250 te

27696250 28981875 ke

28981875 30652500 se

30652500 31745000 te

31745000 33030000 w

33030000 34186875 ye

34186875 35279375 w

35343750 37143125 ha

37143125 38556875 ix

38556875 39906250 tx

39970625 40998750 re

40998750 42091250 t

42091250 43955000 sil

4.2.1.4 Creating the Transcription Files

After creating a label on the recorded data, HTK requires the transcription file to be prepared and stored in a separate file. To train a set of HMMs, every file of training data must have both an associated phone level and word level transcriptions side by side.

The starting point for both sets transcription is an orthographic transcription in HTK label format. The complete set of the orthographic transcription is stored in a single file called Master Label File (MLF).

This Master Label Files (MLFs) may be observed as an index files holding pointers that are pointing to the actual label files which can either be embedded in the same index file or stored elsewhere in the file system.

The Master Label files (MLFs) built in this experiment has the following format.

#!MLF!#

"sample0001.lab"

beadiis

abeba

ketema

yetekesetew

yewha

ixtxixret

.

"sample0002.lab"

bixzu

rixqet

beixgracew

teguzew

yemiiyagenxutixn

wixha

.

Once the word level MLF has been created as shown above, the syllable level MLF can be generated from the already created pronunciation dictionary and the above word level MLF using the label editor HLED. Assuming that the above word level transcription file is named all_mlf, the command

HLED -l '*' -d dict -i syllable.mlf mkphone0.led all_mlf

generates a syllable level transcription of the following form and stores the output in the file called syllable.mlf.

#!MLF!#	te	"'*'/sample0002	gu
"'*'/sample0001	ke	.lab"	ze
.lab"	se	sil	w
sil	te	bix	sp
be	w	zu	ye
a	sp	sp	mii
dii	ye	rix	ya
s	w	qe	ge
sp	ha	t	nxu
a	sp	sp	tix
be	ix	be	n
ba	txix	ix	sp
sp	re	g	wix
ke	t	ra	ha
te	sp	ce	sp
ma	sil	w	sil
sp	.	sp	.
ye		te	

The HLED edit script mkphones0.led contains the commands EX, IS sil sil, DE sp Where EX command replaces each word in the word level.mlf by the corresponding pronunciation in the dictionary file while IS command inserts a silence model sil at the start and end of every utterance. Finally, the delete DE command deletes all short pause "sp" labels, which are not any more wanted in the transcription labels.

4.2.1.5 Coding the Data

Coding the data is the final stage of data preparation, which parameterizes the raw speech waveforms into sequences of feature vectors that are supported by many variants. Among these variants, Fast Fourier Transform (FFT) based and LPC-based analyses are supported by HTK. In this research we use one of the many variant formats that are supported by the toolcalled the Mel Frequency Cepstral Coefficients (MFCCs), which is derived from FFT-based.

Coding can be performed using the tool HCopy configured to automatically convert its input into MFCC vectors from one of the many variants. It requires different parameter, one script file[4] listing the path of the wave file (source file) names along with the destination path which enables to create the .mcf file in parallel as its parameters.

The script file has the following format:

.../sample0001.wav	.../sample0001.mfc
.../sample0002.wav	.../sample0002.mfc
.../sample0003.wav	.../sample0003.mfc

In addition to this, a configuration file (config) is needed which specifies all of the conversion parameters with a reasonable and default settings. These settings are attached in appendix C in which the detail can be found in HTK book [3]. This configuration file

[4] Script file is the file containing all the source and destination address that will enables you to convert the entire .wav file format to the computable .mcf file format that can be used by the tool HTK having an extension of .scp.

reduces the amount of pre-processing required during the training phase, which is a time consuming task.

The .mcf for the entire wave file is created by invoking the command assuming that the above script is stored in the file codetr.scp and config consist of the configuration parameter for MFCCs as shown in appendix C

```
HCopy -T 1 -C config -S codetr.scp
```

4.4.2 Training

In the data preparation stage of the experiment, all the activities that are required for the training of Amharic speech recognizer prepared which is an important step for the success of the well trained data for the recognition. Here, the tasks performed in developing the recognizer by improving the performance of the training through successive refining of the training model to put plain in words exhaustively.

4.4.2.1 HMM Prototype

The first step in HMM training is to define a prototype model. The parameters of this model are not important as they are modified later during training of the model even if it helps to define the model to be used on the later stage.

In this section, the construction of a well trained and recommended set of single-Gaussian monophone HMMs is described. The starting point has a set of identical monophone in which every mean and variance is identical and entire means and variances of the states in the models are simply assigned a value of 0 and 1 respectively.

These are then retrained, short-pause models are added and the silence model is extended slightly and monophones are retrained further.

As stated by Young et al. [52], a good topology for phone-based system is 3-state left-right topology. The same 3-state left-right model is tested for the syllable based recognizer and resulted in degraded performance. Consequently, a 5-state left-right

topology has been used for each syllable with two non-emitting states. A better performance obtained as compared to that of 3-state left to right. These are further proceeded to improve the performance up to 8-state left-to-right tested and better recognition results are obtained with 10 numbers of states in the previous research work. So that, 8-state left-to-right is used prototype is selected for this work and the reason for selecting 8-state topology is, due to the better performance improvement and the initial selected using the standard that are recommended by HTK book [52]. The selected model along with the detail information attached as an Appendix D.

Once the model is selected, the HTK tool HCompV scan a set of data files, compute the global mean and variance and set all of the Gaussians in a given HMM to have the same mean and variance by invoking the following command.

HCompV -C config –T 1 –A -D –m –M hmm0 -f 0.01 -m -S train.scp -M hmm0/proto

Where –C calculate cluster-based mean and variance estimate and store results in the specified directory, -M stores output HMM macro model files in the directory hmm0, -D there will be no warning even if the name of a configuration variable is miss typed the variable will simply be ignored and training data paths (scripts) are stored in the file train.scp in similar fashion to that of the “codetr.scp” without the source data.

Then the command proto create a new version of proto and “vFloors” in the directory hmm0. The previous state in which all the zeros mean and unit variances is replaced by the global speech means and variances.

In addition to this, the “macros” file constructed by appending “vFloors” on the bottom of the global mean and variance which serve as an input in determining the next state. Alternatively, the “macros” file can be constructed using the UNIX command line environment.

cat hmm0/proto hmm0/vFloors > hmm0/macros

Where proto and vFloors are the file created in the directory hmm0 previously and the "macros" file is created by merging the file in the same directory.

Once the global means and variances computed, the new prototype model creates proto and macro file under the directory hmm0. Then, Master Macro File (MMF) called "hmmdefs" constructed from the newly computed model and the monophones that containing a copy for each of the required monophone including "sil".

These are created manually by combining the monophones and the proto file called "hmmdefs" under the hmm0 directory because the initial monophones generated does not contain "sil" phone.

To create the "hmmdefs", copy the content of monophones0 and add the sil at the end of the file hmm0 folder by renaming the monophones0 file to "hmmdefs". And for each phone in "hmmdefs" put the phone in double quotes, then add '~h ' before the phone, finally copy from <BEGINHMM> to <ENDHMM> of the hmm0/proto file and paste it after each phone.

Once you create the monophones the Flat Start Monophones are re-estimated using the HERest tool. The purpose of this is to load all the models in the hmm0 folder, and re-estimate them using the MFCC files listed in the train.scp script, and create a new model set in hmm1.

Then the flat start monophones stored in the directory hmm0 are re-estimated using the embedded re-estimation tool HERest invoked as follows

```
HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0 \ -S train.scp -H
hmm0/macros -H hmm0/hmmdefs -M hmm1 monophones0
```

The effect of this is to load all the models in hmm0 which are listed in the model list monophones0. These are then re-estimated using the data listed in train.scp and the new model set is stored in the directory hmm1. Then the next step is to embed the training using the HEREST command. This simultaneously updates all the HMMs in a system

using all of the training data and the process is repeated two more times, creating new model sets in hmm2 and hmm3.

Each time HEREST is run, it performs a single Baum-Welch re-estimation of the whole set of HMM syllable models simultaneously, and this does not include the short pause “sp” and on the next stage of the training includes the model “sp” in “hmmdefs” of the hmm4 directory.

4.4.2.2 Fixing the Silence Models

According to Rabiner[41], speech recognition system often distinct models for silence and short pauses. The silence model “sil” may have the normal 3-state topology whereas a short pause model “sp” may have just a single state.

So far, the step created HMM models does not include "sp" (short pause) silence model which refers to the types of short pauses that occur between words in normal speech. In order to make the model more robust by allowing individual states to absorb the various impulsive noises in the training data, extra transitions from states 2 to 4 and from states 4 to 2 are added in the silence model.

This is normally done by copying the centre state from the “sil” model in the “hmmdefs” file and adding it to the “sp” model, and then running a special tool called HHED to “tie” the “sp” model to the “sil” model so that they share the same centre state.

Then, the “sp” has its emitting state tied to the centre state of the silence model. The required topology of the two silence models is shown in Fig. 4.3 below

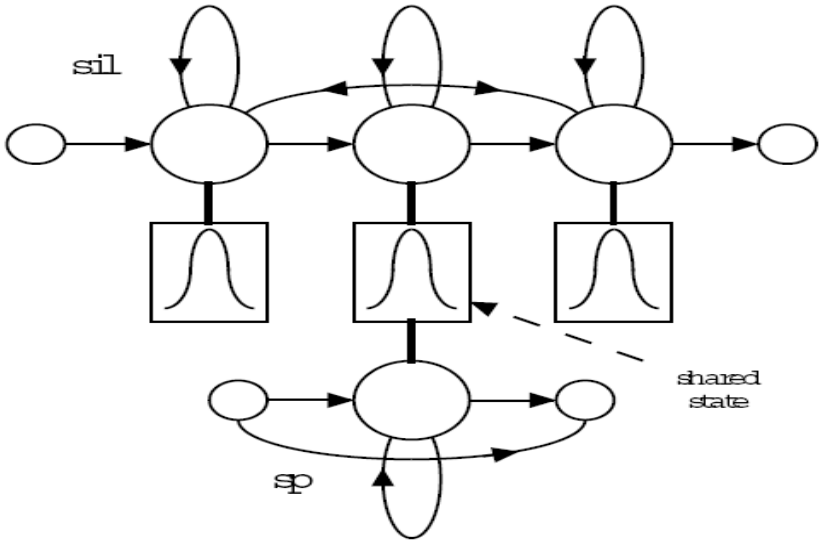

Fig. 4.3: Fixing Silence Models

The silence model "sp" is created using the following procedure in hmm4/hmmdefs as follows:

- Copy and paste the “sil” model of hmm3/hmmdefs and rename the new one “sp” without deleting the old "sil" model, make a copy of the sil model because we need this model to tie in a head of time.

- Remove state 2 and 4 from new “sp” model keeping the centre state of old “sil” model in new “sp” model. So that, you are creating single state short pause sp model so as to avoid the “sil” and “sp” models competing with each other. Beneath these steps, the “sp” model state can be tied to the center state of the “sil” model as shown in figure 4.2.

The HMM command HHEd is used to add the extra transitions required and tie the “sp” state to the centre “sil” state as shown below

HHEd -A -D -T 1 -H hmm4/macros -H hmm4/hmmdefs -M hmm5 sil.hed monophones1

Where sil.hed contains the following commands

AT 2 4 0.2 {sil.transP}

AT 4 2 0.2 {sil.transP}

AT 1 3 0.3 {sp.transP}

TI silst {sil.state[3], sp.state[2]}

The AT commands add transitions to the given transition matrices and the final TI command create a tied-state called silst. The parameters of this tied-state are stored in the "hmmdefs" file and within each silence model, the original state parameters are replaced by the name of this macro.

Finally two more passes of HERest was made using the monophones1 rather than monophones0 which include the short pause "sp" in addition to the "sil" resulting in the final monophone based recognizer.

So that, in the next state a new and updated "hmmdefs" and "macros" file created under the directory hmm6 and hmm7 by invoking the command as below for further training the model

HERest -A -D -T 1 -C config -I phones0.mlf -t 250.0 150.0 3000.0 -S train.scp -H hmm5/macros -H hmm5/hmmdefs -M hmm6 monophones1

4.4.2.3 Realigning the Training Data

As dictionary may contain multiple pronunciations for some words, particularly function words, the phone models created so far used to realign the training data and create new transcriptions. This is done with a single invocation of the HTK recognition tool HVite as shown below

HVite -l '*' -o SWT -b silence -C config -a -H hmm7/macros \ -H hmm7/hmmdefs -i aligned.mlf -m -t 250.0 -y lab \ -I allmlf -S train.scp dict monophones1

This command uses the HMMs stored in hmm7 to transform the input word level transcription ‘allmlf’ 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 of creating the transcription files is that, the recognizer considers all pronunciations for each word and outputs the pronunciation that best matches the acoustic data.

Once the new phone alignments have been created, another two passes of HERest applied to re-estimate the HMM set parameters again. This leaves the set of monophones HMMs created so far in the directory hmm9.

The monophone models created in hmm9 could actually be used for speech recognition, but recognition accuracy can be greatly improved by using Tied-State triphones. The reason for this is that, the Tied-State triphones considers the transition from the previous state to the current state and from the current state to the next state for the phonemes if the phonemes are not starting or final point in a given word.

In addition to this, one of the major problems to be faced in the process of building any HMM based system is the lack of sufficient training data. In order triumph over the problem, HTK provides a variety of mechanism to implement the HMM parameters to be tied together so that the training data is shared and more robust estimates result. So that, the next section focus in discussing the process involved keep away from the problem that aforementioned.

4.4.2.4 Making Triphones from Monophone

Triphones reduce the possibility of error caused by confusing one sound with another, because we are now looking for a distinct sequence of 3 sounds.

To generate a triphone (i.e. a group of 3 phones) declaration from monophones, the "L" phone (i.e. the left-hand phone) proceeds "X" phone and the "R" phone (i.e. the right-hand phone) follows it. Then the final triphone is declared in the form of "L-X+R".

With a triphone model, we are essentially looking for a monophone in the context of the other monophones.

> i.e. The one immediately before and after if they exist, which might be the beginning or end of the words to greatly improve the accuracy of recognition.

Given a set of monophone HMMs, the final stage of model building is to create context dependent triphone HMMs. This is done in two steps. Firstly, the monophone transcriptions are converted to triphone transcriptions and a set of triphone models are created by copying the monophones and re-estimating. Secondly, similar acoustic states of these triphones are tied to ensure that all state distributions of the triphones strongly estimated.

4.4.2.5 Triphone Construction

Triphone Construction is done automatically using HLEd to convert the monophone transcriptions to the desired equivalent set of triphone transcriptions. To do this, 'aligned.mlf' file that is created during training data realignment are used along with 'mktri.led' script as a parameter.

The construction of the triphones is obtained by invoking the following command which generates triphones1 and wintri.mlf:

```
HLEd -A -D -T 1 -n triphones1 -l '*' -i wintri.mlf mktri.led aligned.mlf
```

Where -D Read a dictionary and use this for expanding labels when the EX command is used, -l specifies the directory to store output label files, -i specifies that the output transcriptions are written to the master label file.

The edit script 'mktri.led' contains the commands: WB sp, WB sil and TC. Both WB in the 'mktri.led' scripts defines sp and sil as word boundary symbols.

As a result of this, all the monophone transcriptions have been converted to an equivalent set of triphone transcriptions and are stored in a file named wintri.mlf (Appendix E), and at the same time a list of triphones is printed to the file named triphones1 to improve the

accuracy of the recognizer using the tied state triphones. The fragments of the triphones that are generated are attached as Appendix F.

Executing 'maketrihed' perl script by taking monophones1 and the triphones1 as parameter will enable us to create the mktri.hed script which is an important for the next step file by:

```
perl perlscript/maketrihed monophones1 triphones1
```

After executing the above mentioned command, the next step is creating the cloning of models that can be done efficiently using the HMM editor HHEd:

```
HHEd -B -H hmm9/macros -H hmm9/hmmdefs -M hmm10 mktri.hed monophones1
```

This will create a new model hmm10/hmmdefs and hmm10/macros.

So far the set of triphones used, did cover only the training data. But here a new list of triphones was required, containing all triphones from all type of data. It was then created using the following command:

```
HDMan -A -D -T 1 -b sp -n fulllist -g global.ded -l flog dict-tri
```

This command generated the new list called fulllist that is expanded to include all the triphones needed for recognition both in the testing as well as training the data. And this new list was required by the AU parameter in the 'tree.hed' script.

Next create a new fulllist1 file and copy the contents of the fulllist and triphones1 into it using the script fixfullist.pl, and executing this command will enable to remove any duplicate entries to recognize all of the new previously unseen triphones in the new list.

```
perl fixfulllist.pl fulllist1 fulllist
```

Finally, and for the last time, the models are re-estimated twice using by invoking the HERest command as follows:

HERest -A -D -T 1 -C config -I wintri.mlf -t 250.0 150.0 3000.0 -S train.scp -H hmm10/macros -H hmm10/hmmdefs -M hmm11 triphones1

The outcome of this stage is a set of triphone HMMs with all triphones in a phone set sharing the same transition matrix. When estimating these models, many of the variances in the output distributions will have been floored since there will be insufficient data associated with many of the states and this problems can be prevail over in the following section.

4.4.2.6 Making Tied-State Triphones

In the triphone construction phase, the amount of training data is not sufficient to obtain a reliable estimate of the model parameters, the overall performance may degrade significantly. It seems therefore necessary to reduce the model size. To triumph over this problem, the most common solution is to share some of the model parameters by tying the state output probability distributions among different HMMs as.

In the previous step, the TI parameter used to explicitly tie all members of a set of transition matrices together. However, the choice of which states to tie requires a bit more subtlety since the performance of the recognizer depends crucially on how accurate the state output distributions capture the statistics of the speech data.

HHED provides two mechanisms which allow states to be clustered and then each cluster tied. The first is data-driven and uses a similarity measure between states. The second uses decision trees and is based on asking questions about the left and right contexts of each triphone. The decision tree attempts to find those contexts which make the largest difference to the acoustics and which should therefore distinguish clusters.

A script for automatically generating the file ‘tree.hed’, ‘mkclscript’ available on the HTK demo is used and part of the ‘tree.hed’ script shown in appendix G.

Decision tree state tying performed by running HHED in the normal way, i.e.

HHEd -A -D -T 1 -H hmm12/macros -H hmm12/hmmdefs -M hmm13 tree.hed triphones1 > log

The log file will include summary statistics which give the total number of physical states remaining and the number of models after compacting.

The effect of the AU parameter is to use the decision trees to recognize all of the new previously unseen triphones in the new list.

Once all state-tying has been completed and new models generated, some models may share exactly the same 3 states and transition matrices that are identical. The CO command is used to compact the model set by finding all identical models and tying them together and producing a new list of models called tiedlist.

One of the advantages of using decision tree clustering is that it allows previously unseen triphones to be recognized. To do this, the trees must be saved and this is done by the ST command. Later if new previously unseen triphones are required, for example in the pronunciation of a new vocabulary item, the existing model set can be reloaded into HHED, the trees reloaded using the LT command and then a new extended list of triphones created using the AU command.

After HHED has completed, the effect of tying can be adjusted up on the requirement. Finally, and for the last time, the models are re-estimated twice using HEREST command as follows.

HERest -A -D -T 1 -T 1 -C config -I wintri.mlf -s stats -t 250.0 150.0 3000.0 -S train.scp -H hmm13/macros -H hmm13/hmmdefs -M hmm14 tiedlist

HERest -A -D -T 1 -T 1 -C config -I wintri.mlf -s stats -t 250.0 150.0 3000.0 -S train.scp -H hmm14/macros -H hmm14/hmmdefs -M hmm15 tiedlist

In training, while estimating the models using the HERest command, two of the mean and variances in the output distributions will have a value of not a number (NaN) value. This is due to the absence off sufficient data available to train the data to the next phase

for those words that exist only once or twice in the training phase of the experiment. Had it been more data for the purpose of the training, the problem could not arise because of the sufficient number of phone for the training.

So that, the presence of not a number 'nan' (not a number) value in the training phase would result not to proceed to the next level of the training. So, to triumph over the problem that arise as a result of these single phone existence in the training of the model shifting of the data made from the training set to the testing set without any overlapping of the data between the training and testing set.

Since all the training completed, the next phase is to measure and analyze the performance of the recognizer the given test set data.

4.4.3 Experiment results

Among the 500 sample sentences that has been recorded and preprocessed as per the requirement of the tool, eighty percent of the recorded data is used for the purpose of training and the remaining twenty percent of the data used for testing the recognizer using the available tools of HTK.

The training is completed and all that remains is to measure the over all performance and setting up some parameters. For measuring the performance of the recognizer, the dictionary has already been constructed, and test data has been recorded and labeled as per the requirement of the tool. Thus, all that is necessary is to run the recognizer and evaluate the results using the HTK analysis tool called HResult.

All the data preparation activities discussed in the section 4.2.1 is done before the text is given to the speech-to-text recognizer. These preprocessing activities are done in similar technique as to that of the training such as transcription, 'test.scp' which holds a list of the coded test (.mfc) files, then each test file recognized and its transcription output to an MLF called 'recout.mlf' by executing the following after constructing the "wordnetwork" from the available grammar as shown in Appendix H.

```
HParse gram wordnetwork
```

```
HVite -H hmm15/macros -H hmm15/hmmdefs -S test.scp -l '*' -i recout.mlf -w
wordnetwork -p 0.0 -s 5.0 dict tiedlist
```

Produce a recognition result recout.mlf as shown in appendix I.

The master level file 'testref.mlf' contains syllable level transcriptions for the test file to be recognized and the actual performance can be determined by running HResults as follows taking "testref.mlf" as an input for evaluating the recognition output as shown in appendix J.

```
HResults -I testref.mlf tiedlist recout.mlf
```

All the results obtained using a tool HResult from testing using different criteria are compared and contrasted in the following section.

The first experiment is conducted by training model with native speaker (those who can speak Amharic as a first language) and testing with non native speaker (for those who can not speak Amharic fluently or as a first language)

For these 400 training sets are used. From these training set 300 of the training data is taken from Bereket [3]. And the remaining training 100 training set are collected from the scratch in a similar fashion to that of the training data. The reason for taking 300 training data is that all the data that has been recorded and labeled are equally distributed and native speaker having one fifth of the total data and training and testing with only these data resulted the researcher to have poor result due to the less number training data, which is logically right due to even less number of the training data.

```
------------------------------Overall Results--------------------------------
SENT: %Correct =71.00 [H=71, S=1, N=100]
WORD: %Correct=77.04, Acc= 76.78 [H=302, D=0, S=6, I=1, N=392]
==============================================================
```

As a result of these, an improvement on recognition result accomplished after changing the total number of the training dataset from 100 to 400 dataset. The result obtained from a total of 100 dataset (80% for training and 20% for testing) 55% of the utterance

recognized along with 59.94 % of word level accuracy. After changing the training and testing set of 400 and 100 respectively, the testing set gives the utterance level statistics shows that of the 71 utterance in total, 100 (71%) of the utterances were recognized and the word level statistics indicates that of the 301 words in total, 392 (76.78%) were recognized correctly

The second testing and evaluation is conducted on the training as well as testing the model with native speaker (those who can speak Amharic as a first language). And the similar thing is pertained on selecting the number of training data set that to be used for the purpose of training the model to get an enhanced recognition result.

```
------------------------------Overall Results----------------------------------
SENT: %Correct =82.00 [H=82, S=2, N=100]
WORD: %Correct= 83.76, Acc= 83.29 [H=361, D=0, S=6, I=2, N=431]
==================================================
```

The total number of training data used for this particular testing is 400 with a testing set of 100 data and the result obtained from this testing show that 82 in total of 100 (82%) sentences were recognized. In addition to utterance level recognition, the word level accuracy result shows that 359 words in total, 431(83.76%) recognized successfully.

The third testing and evaluation conducted on the training as well as testing the model with non native speaker (those who can't speak Amharic as a first language). And for these additional 100 samples are recorded and labeled to have a uniform training as well as testing set for this paper.

```
------------------------------Overall Results----------------------------------
SENT: %Correct =67.00 [H=67, S=2, N=100]
WORD: %Correct= 69.13, Acc= 69.13 [H=271, D=0, S=5, I=0, N=392]
==================================================
```

To conduct the recognition 400 training set are used with a total testing set of 100 data and the result obtained from this testing show that 67 in total of 100 (67%) sentences

recognized successfully with a word level accuracy result in that 271 words in total, 392(69.13%) recognized successfully.

The fourth and the last testing and evaluation is conducted on the training the model with non native speaker and testing with native speaker are taken and the following result were achieved.

----------Overall Results----------

SENT: %Correct =75.00 [H=75, S=3, N=100]

WORD: %Correct= 79.58, Acc= 79.58 [H=343, D=0, S=2, I=8, N=431]

==========

The result obtained from this testing set the utterance level statistics shows that 75 utterances in total, 100 (75%) of the utterances were recognized and the word level statistics accuracy indicates that of the 335 words in total, 431 (77.72%) recognized correctly.

The table 4.1 shows that the statistics obtained from the recognition result of the above four experiment.

	Native with Native		Native with Non-Native		Non-Native with Non-Native		Non-Native with Native	
	Sent	word	Sent	word	Sent	word	Sent	word
Correct labels (H)	82	359	71	301	67	271	75	335
Deletions (D)		0		0		1		2
Substitutions (S)	1	6	4	6	2	5	3	2
Insertions (I)		2		1		0		8
Utterances (N)	100	431	100	392	100	392	100	431

Table 4.1 recognition result statistics

For the above experiment conducted, the development test set used 100 utterances with a total training set of 400 sentences and the overall recognition results are based on the calculation that are provided in equation (1) and (2) shown below for the clarification.

The percentage number of labels correctly recognized is given by

Percentage is calculated as

$$\%\text{Correct} = {}^{H}\!/_{N} * 100\% \quad (1)$$

And the accuracy is computed by:

$$\%\text{Accuracy} = {}^{(H-I)}\!/_{N} * 100\% \quad (2)$$

Where H- is the number of correct labels (Hits), D- is the number of deletions, and S- is the number of substitutions, I- is the number of insertions and N- is the total number of utterances.

The sentence recognized notates the sentence-level accuracy based on the total number of label files which are identical to the transcription files and the word recognized measures the word level accuracy based on the matches between the label files and the transcriptions.

However, all these results are based on the development of the test set that compromise. The entire dataset were used over and over again in order to test the factual robustness of the system that recognizes the Amharic speech without any common training sentences in the testing set and vice verse.

The following table 4.1 presents summary of the recognition result and the next section presents the over all analysis of the result obtained while attempting to investigate a general approach for Amharic speech-to-text recognition.

	Training set		**Testing set**		**Result**		
	Native speaker	**Non-Native speaker**	**Native speaker**	**Non Native speaker**	**Sentence recognized (%)**	**Word Recognized (%)**	**Word level Accuracy (%)**
1	yes	no	yes	no	82	83.76	83.29
2	yes	no	no	yes	71	77.04	76.78
3	no	yes	no	yes	67	69.13	69.13
4	no	yes	yes	no	75	79.58	77.72

Table 4.2 experimental result

4.4.4 Analysis of the result

A total of twenty speaker of the language were selected to train and evaluate the system. Half of them are female and the remaining are male speaker. Even though, male and female are not treated separately, rather the speaker is treated based on their ability to speak Amharic language as a mother tongue or not. Obviously, we find an individual that can speak Amharic fluently as a first language being not Amharic a mother tongue.

For this work people are selected from those who can not speak Amharic as a first language with determination of their capability to speak out as a native and non native speaker from Oromo, Tigre, Gurage, Hadya and Amhara with equal distribution.

In general perfect 100% accuracy level result is unattainable due to the a number of reason such as: the environment of the sound recording, the microphone were also susceptible to external interference, unavailability of large corpus for the training along with the inaccessibility of good support in the linguistic representation specially for those language that are not technologically supported like Amharic and the faultless of the speaker to speak the language Amharic as a first language.

As we can see from the table 4.2, training the data with a native speaker and testing with a non-native Amharic speaker resulted in 71 of the sentences recognized out of the total 100 sentences having 76.78% word level accuracy.

While the second experiment, training and testing with a native speaker resulted in 82% sentence having 83.29% word level accuracy which is better recognition result. These results were to be expected, since the data set was recorded in a similar environment and there exist a difference in the capability to utter the sentence from Native to Non-Native speaker of the language Amharic.

Further, the recognition results degrades tremendously as we move away from the native to Non-Native speaker provided the same recording environment as to the other experiment. The experiment result in Non-native to non-native shows that, 67% of the sentences recognized with the word level accuracy 69.13% due to the same reason above mentioned.

Where as, the other recognition result obtained from training with non-native and testing with native speaker of the Amharic shows that 75% of the sentences were recognized with the word level accuracy of 77.72% percent which is a better result as compare to that of the non-native to non-native testing and relatively similar to that of native to non-native recognition result.

As to the belief of the researcher and from real life situation, a word or sentences uttered by non-native speaker can be understood better by native speaker of the language than that of the reverse given the same environment. In addition, this experiment result could perhaps improve to a higher percentage level both for the word and sentence level had there been more data both for the training and testing the recognizer. Further more, the percentage of recognition result for both testing with native and non-native could also be improved if the entire speech were recorded from the floor rather than using the secondary data as to that of the remaining experiment.

From these experimental results, one can observe that a better recognition result is obtained in training and testing with a native speaker of the language Amharic. Next to this, a better recognition result achieved from training with non-native and testing with native speaker and similarly the next better recognition result is obtained from training with native speaker and testing with non-native speaker. Finally, the lowest speech recognition result is accomplished from the training and testing with a non-native speaker of the language.

So this shows that, there exists a high degree of deviation as we move away from the native to non native. Provided the constraint of recording environment, external interference of speaker and relatively small side data, the performance measure of the recognition result obtained in this experiment is nearly good in terms of quality, simplicity and naturalness.

CHAPTER FIVE

Conclusions and Recommendations

5.1 Introduction

This section present the conclusions drawn from the findings of the experiment and the recommendations on further actions that can be taken and future research pertaining to speech recognition are recommended.

5.2 Conclusions

Speech recognition has been developed gradually over the last decades and it has been incorporated into several applications area such as for people with hard of hearing and visual impaired, educational applications, telecommunications and multimedia service, man-to-machine communication, fundamental of applied research, dictation systems command and control system. Especially, people with disability are the suffering from lack of communication with their relatives.

As a result of this and difficulties stated in the statement of the problem, many researches are emerging as a result of these problem and others that are discussed in and the opportunities that hinders their inability to communicate with other. Some of the problems are solved by different researcher through different techniques and this work appeared in favor of supporting the area specifically to the local language focusing on native and non-native speaker of the language Amharic.

The Hidden Markov Model (HMM) is selected along with the Hidden Markov Toolkit (HTK) for developing a general approach that can recognize and convert an Amharic SPEECH-TO-TEXT.

To conduct this research sample text has been selected and further processed; speech file recorded dividing in training and testing after processing the dataset as per the requirement of the tool HTKs. Then, the selected training data are trained using the appropriate tool and triphone is constructed to get better recognition result is out of it.

In this experiment, 82% of recognition of sentences having 83.29% word level accuracy, 71% sentence recognition having 76.78% word level accuracy, 75% recognition having 77.72% word level accuracy, and 67% sentence recognition having 69.13% word level accuracy achieved for native with native, Native with non-native, Non-native with native and non-native with non-native respectively.

Then, the recognition results compared and contrasted after evaluating the model despite the circumstances and limitations that defend against the existence of this work.

In the presence of frequent power interruption, lack of Amharic pronunciation dictionary, inaccessibility of room for recording the corpus a large, sufficient and a well balanced reference database, the results obtained are encouraging. And had there been more time, the performance of recognizer could be improved to the most level.

Finally, provided the constraint, the result obtained is promising and serve as a proof that it is possible to build general speech recognition technique that convert an Amharic SPEECH-TO-TEXT using the toolkit HTK and the HMM modeling technique.

Based on the experience and different problems that are raised in this experiment, the following recommendations are drawn and forwarded to point out the upcoming research in the speech related area.

5.3 Recommendations

The main aim of this work is to investigate the general approach for the speech recognition technique that is capable of converting an Amharic SPEECH-TO-TEXT.

Amharic is one of the Semitic languages through which the continuation with effort to build recognizer for the languages that are not technologically supported even though the language are traced back to the first millennium B.C

- One of the major problems that faced in the experiment is absence of well defined pronunciation dictionary for the language Amharic that consist all the word. So that, the construction of dictionary might be of much service for feature research.

- In the experimentation, utterances made by people whose first language are Amharic, Oromiffa, Tigray, Hadya and Gurage with a total sample of 500 utterances. So that, the recognizer developed out of it might not be well serving for all human kind in the world except that their mother tongue is Amharic so that it should be extended for the support for those who can not speaks out Amharic as a first language.

- Since this research mainly focus for those native and non native speakers of the language Amharic without considering the dialectical variations among the non native speaker of the language Amharic. So that, this can further research need to explore for dialectical variations among the non native speaker of the language.

- In addition to the statistical approach of the Hidden Markov model (HMM) speech recognition can be extended using the Artificial Neural Network (ANN) approach or a combination of ANN and HMM for recognizing speech the speech.

Reference

1. Alan W and Kevin A. (2003): *Building Synthetic Voices for FestVox 2.0* available at: http://www.festvox.org/bsv/bsv.pdf (last accessed date march 2009)

2. Antonio P. and Jos S. (2006): *Speech Recognition over Digital Channels*. John Wiley and sons, Ltd

3. Bereket K. (2008): developing a speech synthesizer for amharic language using Hidden Markov Model, M.Sc. Thesis, Addis Ababa University, Addis Ababa

4. Brendan M.: *What is Phonetics*? available at: http://www.wisegeek.com/what-is-phonetics.htm (last accessed date march 2009)

5. *Child of the World : Amharic* available at http://ourworld.compuserve.com/homepages/GenX_jt_mtjr/GenXAmharic.htm (last accessed date march 2009)

6. Daniel J. & James H. Martin (2006): *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*. University of Colorado

7. David G. : *Audio Signal Classification: An Overview*, School of Computing Science Simon Fraser University Burnaby,

8. Davis, K., Biddulph R. & Balashek, S. (1952): *Automatic recognition of spoken digits*. *The Journal of the Acoustic Society of America*, 24(6), 637-642. Available at source http://www.nexus.carleton.ca/~kekoura/history.html (last accessed Nov 08)

9. Dawit Y. (1998): *Applying Interface Agent technology to selective dissemination of Information user profile management: the case of ILRI ALERT:* M.Sc. Thesis. Addis Ababa University

10. Deller, J., Hansen, J., and Proakis, J. (1999) *Discrete-Time Processing of Speech Signals*, 2nd Edition. Wiley-IEEE Press.

11. Digital Signal Processing: available at: http://www.dsptutor.freeuk.com/index.htm. (Last accessed April 9,2009)

12. Douglas O. (2003): Interacting with Computers by Voice: Automatic Speech Recognition and Synthesis, IEEE (91 9)

13. Ethnologue004.Languages of the World, 14th Edition. http://www.ethnologue.com/14/showlanguage.asp?code=AMH (last accessed date march 2009)

14. Forester W. Isen 2008: *Fundamentals of Discrete Signal Processing*

15. Francois D. (2005) How Speech Recognition Works http://www.gignews.com/fdlspeech1.htm

16. Getahun A. (1995) : **ዘመናዊ የአማርኛ ሰዋስው በቀላል አቀራረብ** :: **ንግድ ማተሚያ ድርጅት**

17. Henock Lulseged (2003). *Concatinative Text-to-Speech synthesis for Amharic Language*. M.Sc. Thesis. Addis Ababa University

18. Ithiopia Series-Amharic, the language of Ethiopia : available at http://www.alumbo.com/article/14552-Ithiopia-Series-Amharic-the-language-of-Ethiopia.html

19. John H., Wendy H. (2001): *Speech Synthesis and Recognition*: Second Edition, Taylor & Francis e-Library.

20. John P. (2006) Automatic Speech Recognition Techniques http://www.globalsecurity.org/intell/systems/asr-tech.htm

21. Jose (2007): Speech to Text: timesaver or time waster? . Available at : http://www.academicproductivity.com/2007/speech-to-text-timesaver-or-time-waster/ (Last accessed May 2009)

22. Juang B. H. and Rabiner L. R.(1991) : Hidden Markov Models for Speech Recognition, Speech Research Department (33 : 3)

23. Kimberlee A. Kemble (nd): *An Introduction to Speech Recognition Program Manager*, Voice Systems Middleware Education IBM Corporation

24. Kinfe T. (2002) : sub-word based Amharic word recognition : an experiment using Hidden Markov Model, M.Sc. Thesis, Addis Ababa University, Addis Ababa

25. Laine B. (1998) *Text-To-Speech synthesis of the Amharic Language.* M.Sc. Thesis. Addis Ababa University: Technology faculty, Addis Ababa

26. Lawrence R. Challenges in Speech Recognition, Rutgers University: available at http://www.msri.org/publications/ln/hosted/nas/2002/rabiner/1/banner/03.html

27. Lawrence R.(1989) :*A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, IEEE (77: 2)

28. Leslau W. (1995) : *Reference Grammer of Amharic*, Wiesbaden Harrassowitz.

29. Magnus K : *Hidden Markov Models* http://www.math.chalmers.se/~olleh/Markov_Karlsson.pdf (last accessed date May 2009)

30. Magnuson T., Blomberg, M (2000). :*Acoustic analysis of dysenteric speech and some implications for automatic speech recognition* (41:1)

31. Marcel D: *Natural Language Processing*, Encyclopedia of Library and Information Science, 2 nd Edition. available at:

http://www.cnlp.org/publications/03NLP.LIS.Encyclopedia.pdf (last accessed date march 09)

32. Mark G, Steve Young (2008): *The Application of Hidden Markov Models in Speech Recognition*, Cambridge University press Available at: http://mi.eng.cam.ac.uk/~mjfg/mjfg_NOW.pdf. (last accessed may 2009)

33. Masaaki H. (2003): *Human speech production mechanism.* Available at http://www.analogue.org/network/Speech%20production.pdf . (last accessed April 2009)

34. Michael C., Tom O., Frank M.: *VOICE IDENTIFICATION: The Aural/ Spectrographic Method*, McDermott Ltd. Owl Investigations, Inc. available at : http://www.owlinvestigations.com/forensic_articles/aural_spectrographic/fulltext.html#spectrograph (last accessed April 2009)

35. Michael G. and Kristian K. (2007): *Robust Speech recognition and understanding*, I-Tech education and Publishing

36. Morka M. (2001): Text-To-Speech for Afaan Oromo: *M.Sc. Thesis. Addis Ababa University: Technology faculty, Addis Ababa*

37. Mustafa N. Kaynak, Qi Zhi, Adrian David Cheok, Kuntal Sengupta, Zhang Jian, and Ko Chi Chung (2004): *Analysis of Lip Geometric Features for Audio-Visual Speech Recognition.* IEEE transaction on system, man, and cybernetics-part A: System and humans. 34(4):564-570

38. Nadew T. (2008) : *Formant based speech synthesis for Amharic vowel* M.Sc. Thesis. Addis Ababa University

39. Narada W. (1996) : http://jedlik.phy.bme.hu/~gerjanos/HMM/node3.html (Last accessed 2009)

40. Phil B. (2004) : *Hidden Markov Models* available at http://www.cs.mu.oz.au/460/2004/materials/hmm-tutorial.pdf (last accessed date May 2009)

41. Rabiner, L.R., and B-H, Juang. *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1993 .

42. Ron C., Victor Z. (1996): *Survey of the State of the Art in Human Language Technology* (unpublished document) Massachusetts Institute of Technology Available from http://cslu.cse.ogi.edu/HLTsurvey/ch1node3.html#SECTION11 (last accessed Nov 2008)

43. Sami L. (1999): *Review of speech synthesis technology*. Master Thesis. Helsinki University of Technology. Available at: http://www.acoustics.hut.fi/publications/files/theses/lemmetty_mst/thesis.pdf . (Last accessed February 2009)

44. Sebsibe H. , S P Kishore , Alan W Black, Rohit Kumar , And Rajeev Sangal : *Unit Selection Voice For Amharic Using Festvox* : (2004) 5th ISCA Speech Synthesis Workshop - Pittsburgh

45. silk : A resource for Norwegian language technology, available at : http://portal.bibliotekivest.no/terminology.htm

46. Solomon B. (2001): *Isolated Amharic Consonant-Vowel (CV) syllable Recognition: an experiment using the Hidden Markov Model,* M.Sc. Thesis, Addis Ababa University, Addis Ababa

47. Solomon T. (2005): *Automatic Speech Recognition for Amharic*, Hamburg University.

48. Solomon T., Martha Y., Wolfgang M. (nd): *Amharic Speech Recognition: Past, Present and Future* University of Hamburg. Available at: http://nlp.amharic.org/Members/solomon/papers/ies16conference.pdf. Last accessed march 2009

49. Stephen C. (2002): *Speech Recognition HOWTO.* Available at: http://www.faqs.org/docs/Linux-HOWTO/Speech-Recognition-HOWTO.html#LEGAL . (Last accessed April 2009)

50. Victor Z, Ron C, and Wayne W (nd). *Speech Recognition.* Available at: http://cslu.cse.ogi.edu/HLTsurvey/ch1node4.html . Last accessed April 2009

51. Waleed H, Nikola K.(2009): *The Concepts of Hidden Markov Model in Speech Recognition*, Technical Report, University of Otago, New Zealand

52. Young S., et al. (2006): The HTK Book. Microsoft Corporation.

53. Zegaye S. (2003) *HMM based large vocabulary, Speaker Independent, Continuous Amharic Speech Recognizer*, M.Sc. Thesis, Addis Ababa University, Addis Ababa

54. http://www.answers.com/topic/natural-language-recognition

55. http://www.wisegeek.com/ what is consonant and vowel (Last accessed April 2009)

56. http://htk.eng.cam.ac.uk/

57. http://en.wikipedia.org/wiki/Speech_to_text

58. http://www.stsn.org/Admins.html#Definitions

59. http://jedlik.phy.bme.hu/~gerjanos/HMM/node11.html#SECTION00243110000000000000

Appendix

Appendix A: the language of Ethiopia (“Fidel”)

Appendix B: Amharic Phonetic List, IPA Equivalence and its ASCII Transliteration Table [37]

IPA	Transcription	Amharic equivalence
Consonants		
[p]	[p]	ፕ
[t]	[t]	ት
[k]	[k]	ክ
[ʔ]	[ax]	ዕ
[b]	[b]	ብ
[d]	[d]	ድ
[g]	[g]	ግ
[pʾ]	[px]	ጵ
[tʾ]	[tx]	ጥ
[cʾ]	[cx]	ጭ
[q]	[q]	ቅ
[f]	[f]	ፍ
[s]	[s]	ስ
[ʃ]	[sx]	ሽ
[h]	[h]	ሀ
[sʾ]	[xx]	ጽ
[tʃ]	[c]	ች
[gʾ]	[j]	ጅ
[m]	[m]	ም
[n]	[n]	ን
[nʾ]	[nx]	ኝ
[l]	[l]	ል
[r]	[r]	ር
[j]	[y]	ይ
[w]	[w]	ው
[v]	[v]	ቭ
[z]	[z]	ዝ
[zʾ]	[zx]	ዥ
Vowels		
[ɛ]	[e]	ኧ
[ʊ]	[u]	ኡ
[ɪ]	[ii]	ኢ
[ɑ]	[a]	ኣ
[e]	[ie]	ኤ
[ɨ]	[ix]	እ
[o]	[o]	ኦ

Appendix C: The configuration parameter: config

```
# Coding parameters

TARGETKIND = MFCC_0

TARGETRATE = 100000.0

SAVECOMPRESSED = T

SAVEWITHCRC = T

WINDOWSIZE = 250000.0

USEHAMMING = T

PREEMCOEF = 0.97

NUMCHANS = 26

CEPLIFTER = 22

NUMCEPS = 12

ENORMALISE = F
```

Appendix D: The Prototype HMM

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"
<BeginHMM>
 <NumStates> 10
 <State> 2
   <Mean> 39
    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 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> 39
    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 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> 39
    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 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> 39
    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 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> 39
     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 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> 39
     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 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> 5
   <Mean> 39
     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 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> 39
     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 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> 6
   <Mean> 39
     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 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> 39
     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 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> 7
   <Mean> 39
     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 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> 39
```

```
    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 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> 8
  <Mean> 39
    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 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> 39
    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 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> 9
  <Mean> 39
    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 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> 39
    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 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> 10
 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 0.0 0.2 0.2 0.2 0.2 0.1 0.1 0.0 0.0 0.0
 0.0 0.0 0.2 0.2 0.2 0.2 0.1 0.1 0.0 0.0
 0.0 0.0 0.0 0.2 0.2 0.2 0.2 0.1 0.1 0.0
 0.0 0.0 0.0 0.0 0.2 0.2 0.2 0.2 0.1 0.1
 0.0 0.1 0.1 0.2 0.2 0.2 0.2 0.0 0.0 0.0
 0.0 0.0 0.1 0.1 0.2 0.2 0.2 0.2 0.0 0.0
 0.0 0.0 0.0 0.1 0.1 0.2 0.2 0.2 0.2 0.0
 0.0 0.0 0.0 0.0 0.1 0.1 0.2 0.2 0.2 0.2
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Appendix E: Triphone construction sample

CL triphones1

TI T_a {(*-a+*,a+*,*-a).transP}

TI T_sp {(*-sp+*,sp+*,*-sp).transP}

TI T_ba {(*-ba+*,ba+*,*-ba).transP}

.

TI T_w {(*-w+*,w+*,*-w).transP}

TI T_tu {(*-tu+*,tu+*,*-tu).transP}

TI T_n {(*-n+*,n+*,*-n).transP}

TI T_nxu {(*-nxu+*,nxu+*,*-nxu).transP}

TI T_txi {(*-txi+*,txi+*,*-txi).transP}

TI T_qie {(*-qie+*,qie+*,*-qie).transP}

TI T_sxe {(*-sxe+*,sxe+*,*-sxe).transP}

.

.

.

TI T_wu {(*-wu+*,wu+*,*-wu).transP}

TI T_txie {(*-txie+*,txie+*,*-txie).transP}

Appendix F: Triphone generated

sil	gix	zo	sa	qix
sp	qii	xxie	ru	yix
be	ro	sie	sii	ku
a	c	txa	wa	bii
dii	l	ja	rie	mie
s	je	ra	rii	f
ba	m	ce	ho	mo
ke	ya	gii	pii	nix
te	de	zie	cxii	xx
ma	q	hu	rix	twa
ye	d	sxa	nx	tie
se	we	cx	gwa	v
w	qa	wo	p	zxix
ha	ta	da	nu	fix
ix	le	wii	fa	nie
tx	na	nii	ko	go
re	cxe	k	mu	nxe
t	la	j	yu	cxa
b	y	su	zii	nxa
zu	txe	xxe	fu	lu
r	hii	fii	nxo	six
qe	tu	ii	ka	mix
g	me	px	kwa	tix
gu	po	yo	qwa	dix
ze	lii	lie	txu	jix
mii	h	lo	xxix	xxu
ge	bu	cix	no	cie
nxu	ga	so	ve	bo
n	to	fe	tii	do
ne	hix	cxo	pa	bix

gie	za	hie	qie	wu
hwa	wix	txix	ju	dwa
qu	sx	sxix	cxix	rwa
kii	ca	qo	swa	vii
bie	kie	nxix	fie	fwa
kix	cu	z	sxe	co
xxa	txwa	yie	du	nwa
jii	lwa	die	sxii	zwa
fo	txie	o	ie	bwa
lix	mwa	zix	txo	cxu

Appendix G: tree.hed

```
RO 100.0 stats
TR 0

QS "L-Vowel" {*^a-*,*^e-*,*^ii-*,*^ix-*,*^ie-*,*^o-*,*^u-*}
QS "L-Stop" {*^b-*,*^d-*,*^px-*,*^g-*,*^k-*,*^p-*,*^t-*,*^tx-*,*^q-*}
QS "L-Nasal"       {*^m-*,*^n-*,*^nx-*}
QS "L-Fricative" {*^f-*,*^xx-*,*^v-*,*^s-*,*^sx-*,*^z-*,*^zx-*}
QS "L-Liquid"      {*^l-*,*^r-*}
                                        •
                                        •
QS "L-Medium_Vowel"          {*^ie-*,*^e-*,*^o-*}
QS "L-Low_Vowel"        {*^a-*}
QS "L-Rounded_Vowel"          {*^o-*,*^u-*}
QS "L-IVowel"               {*^ix-*,*^ii-*,*^ie-*}
QS "L-EVowel"                {*^e-*}
QS "L-AVowel"                {*^a-*}
QS "L-OVowel"                {*^o-*}
QS "L-UVowel"                {*^u-*}
QS "L-Unvoiced_Consonant"      {*^c-*,*^f-*,*^k-*,*^p-*,*^s-*,*^sx-*,*^t-*,*^tx-*}
                                        •
                                        •
QS "L-xx"           {*^xx-*}
QS "L-c"           {*^c-*}
QS "L-j"           {*^j-*}
QS "L-m"           {*^m-*}
QS "L-n"           {*^n-*}
QS "L-nx"          {*^nx-*}
QS "L-l"           {*^l-*}
QS "L-r"           {*^r-*}
QS "L-ii"          {*^ii-*}
QS "L-a"           {*^a-*}
```

```
QS "L-ie"          {*^ie-*}
QS "L-ix"          {*^ix-*}
QS "L-o"           {*^o-*}
QS "L-pau"          {*^pau-*}
QS "L-SIL"          {*^SIL-*}
QS "L-h#"          {*^h#-*}
QS "L-brth"         {*^brth-*}
TR 2

TB 350 "ST_a_2_" {("a","*-a+*","a+*","*-a").state[2]}
TB 350 "ST_ba_2_" {("ba","*-ba+*","ba+*","*-ba").state[2]}
TB 350 "ST_l_2_" {("l","*-l+*","l+*","*-l").state[2]}
TB 350 "ST_ta_2_" {("ta","*-ta+*","ta+*","*-ta").state[2]}
TB 350 "ST_ce_2_" {("ce","*-ce+*","ce+*","*-ce").state[2]}
TB 350 "ST_w_2_" {("w","*-w+*","w+*","*-w").state[2]}
                                •
                                •
                                •
TB 350 "ST_txwa_4_" {("txwa","*-txwa+*","txwa+*","*-txwa").state[4]}
TB 350 "ST_v_4_" {("v","*-v+*","v+*","*-v").state[4]}
TB 350 "ST_zxix_4_" {("zxix","*-zxix+*","zxix+*","*-zxix").state[4]}
TB 350 "ST_wu_4_" {("wu","*-wu+*","wu+*","*-wu").state[4]}
TB 350 "ST_txie_4_" {("txie","*-txie+*","txie+*","*-txie").state[4]}
TB 350 "ST_qie_4_" {("qie","*-qie+*","qie+*","*-qie").state[4]}
TB 350 "ST_mwa_4_" {("mwa","*-mwa+*","mwa+*","*-mwa").state[4]}
TB 350 "ST_sil_4_" {("sil","*-sil+*","sil+*","*-sil").state[4]}
TR 1
AU "fulllist"
CO "tiedlist"

ST "trees"
```

Appendix H "wordnetwork"

```
VERSION=1.0
N=374  L=921
I=0    W=SENT-END
I=1    W=sil
I=2    W=!NULL
I=3    W=go
I=4    W=mwa
I=5    W=ju
I=6    W=nxu
I=7    W=cxix
I=8    W=du
I=9    W=xxe
I=10   W=qie
I=11   W=txie
I=12   W=zxix
I=13   W=v
I=14   W=txwa
I=15   W=sx


I=353  W=de
I=354  W=s
I=355  W=dii
I=356  W=da
I=357  W=n
I=358  W=nxa
I=359  W=za
I=360  W=t
I=361  W=yo
I=362  W=ro
I=363  W=b
I=364  W=be
I=365  W=w
I=366  W=ce
I=367  W=ta
I=368  W=ba
I=369  W=a
I=370  W=DIAL
I=371  W=SENT-START
I=372  W=!NULL
I=373  W=!NULL


J=0    S=2    E=0
J=1    S=187  E=0
J=2    S=185  E=1
J=3    S=1    E=2
J=4    S=3    E=2
J=5    S=4    E=2
J=6    S=5    E=2
J=7    S=6    E=2
J=8    S=7    E=2
J=9    S=8    E=2
J=10   S=9    E=2
J=11   S=10   E=2
J=12   S=11   E=2
J=13   S=12   E=2
J=14   S=13   E=2
J=15   S=14   E=2
J=16   S=15   E=2
J=17   S=16   E=2
J=18   S=17   E=2
J=19   S=18   E=2
J=20   S=19   E=2
J=21   S=20   E=2
J=22   S=21   E=2
J=23   S=22   E=2
J=24   S=23   E=2
J=25   S=24   E=2
J=26   S=25   E=2
J=27   S=26   E=2
J=28   S=27   E=2
J=29   S=28   E=2
J=30   S=29   E=2
J=31   S=30   E=2
J=32   S=31   E=2
J=33   S=32   E=2
```

Appendix I sample recognition output

```
#!MLF!#
"'*'/sample0161.rec"
12600000 13400000 ke -399.773712
13400000 14300000 k -649.181335
14300000 15100000 nxa -612.041931
15100000 16100000 me -795.159119
16100000 17600000 la -1191.398682
17600000 18500000 sii -704.109680
18500000 19000000 txe -352.604980
19000000 19500000 c -410.366852
19500000 20200000 zu -536.431580
20200000 20400000 l -158.708755
20400000 21100000 be -376.760254
21100000 21500000 tix -294.373383
21500000 22700000 ye -867.937195
22700000 23500000 gu -526.638855
23500000 23800000 ke -167.529892
23800000 24100000 tix -210.380356
24100000 24500000 ze -270.724182
24500000 25200000 gu -455.115051
25200000 28000000 sil -786.409546
28000000 28200000 ii -108.159096
28200000 29500000 ye -826.531311
29500000 30200000 te -522.625488
30200000 30900000 w -540.291260
30900000 31800000 q -744.315186
31800000 32500000 sa -549.529907
32500000 32900000 ke -200.801498
32900000 33300000 te -269.794495
33300000 33600000 ze -217.312347
33600000 33900000 tu -173.949036
33900000 34700000 te -599.175598
34700000 36300000 ma -1137.409912
36300000 36600000 rii -213.306564
36600000 36900000 ze -205.455948
36900000 37100000 fe -146.503342
37100000 37800000 c -501.246338
37800000 38000000 ca -120.628464
38000000 39000000 c -766.913574
39000000 39800000 gix -517.560181
39800000 40600000 nxa -586.727295
40600000 41400000 wa -467.685791
41400000 42500000 tx -716.528503
42500000 43100000 ru -455.939240
43100000 44200000 be -484.321808
44200000 44700000 da -361.279541
44700000 45100000 sa -323.752625
45100000 45400000 q -251.132172
45400000 47400000 t -1186.151001
.
"'*'/sample0162.rec"
14000000 14200000 lie -96.782745
14200000 15200000 ix -716.873413
15200000 17100000 ze -1289.521118
17100000 17800000 ru -467.742523
17800000 18200000 gii -300.457214
18200000 18900000 n -528.572876
18900000 19100000 nie -142.711426
19100000 19900000 zie -507.118317
19900000 21100000 s -840.098450
21100000 21900000 nxa -565.395874
21900000 22500000 de -427.420471
22500000 23200000 qa -524.071472
23200000 23900000 la -566.081665
23900000 24800000 ye -576.828247
24800000 25000000 le -166.236603
25000000 25500000 ya -422.465942
25500000 25700000 we -142.032883
25700000 26400000 dix -286.365662
26400000 26700000 be -174.527740
26700000 27100000 to -312.207916
27100000 27300000 qa -148.902390
27300000 27500000 ge -143.283737
27500000 29100000 d -1065.606201
29100000 29600000 pii -378.401459
29600000 29900000 q -243.778870
29900000 30500000 la -447.661011
30500000 30700000 to -185.576111
30700000 31500000 c -668.994629
31500000 32000000 ke -240.990875
32000000 32600000 te -451.682800
32600000 34300000 ze -1139.530273
34300000 35400000 re -790.063232
35400000 35700000 dix -168.081772
35700000 36000000 sil -146.962112
36000000 36200000 dix -105.110641
36200000 37600000 t -782.414978
```

Appendix J sample label file

8554375 13545000 sil
13623750 14891250 ke
14891250 16158750 me
16158750 17188125 la
17188125 18138750 w
18138750 19326875 ii
19326875 21070000 t
21148750 23366875 yo
23366875 24634375 px
24713750 26060000 ya
26060000 28198750 sil
28198750 29783125 ye
29783125 30733750 te

30812500 31683750 w
31683750 33030625 txa
33030625 34456250 txu
34456250 35565625 te
35565625 36595000 ma
36595000 37466250 rii
37466250 38496250 wo
38496250 39208750 c
39208750 40476250 ye
40476250 41585625 mii
41585625 42694375 ma
42694375 44278750 ru
44278750 45704375 be
45704375 46971875 t
47050625 52278750 sil