
**ADDIS ABABA UNIVERSITY**

**COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES**

**SCHOOL OF INFORMATION SCIENCE**

# A LARGE VOCABULARY, SPEAKER-INDEPENDENT, CONTINUOUS SPEECH RECOGNITION SYSTEM FOR AFAAN OROMO: USING BROADCAST NEWS SPEECH CORPUS

**By**

**YADETA GONFA GUTU**

A THESIS SUBMITTED TO THE SCHOOL OF INFORMATION SCIENCE OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN INFORMATION SCIENCE

**OCTOBER, 2016**

**ADDIS ABABA, ETHIOPIA**

**ADDIS ABABA UNIVERSITY**

**COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES**

**SCHOOL OF INFORMATION SCIENCE**

# A LARGE VOCABULARY, SPEAKER-INDEPENDENT, CONTINUOUS SPEECH RECOGNITION SYSTEM FOR AFAAN OROMO: USING BROADCAST NEWS SPEECH CORPUS

**BY: YADETA GONFA GUTU**

**OCTOBER, 2016**

**Name and signature of members of the examining board for Approval**

| Name | Signature |
|---|---|
| 1. Dr. MARTHA YIFIRU (Advisor) | _____ |
| 2. Dr. SOLOMON TEFERRA (Examiner) | _____ |
| 3. Dr. RAHEL BEKELE (Examiner) | _____ |
| 4. _____ | _____ |

## DEDICATION

**This work is dedicated to my wife Alemnesh Temesgen.**

## DECLARATION

**This thesis is my original work and it has not been submitted for a degree in any other University.**

**_____**

**YADETA GONFA GUTU**

**OCTOBER, 2016**

**The thesis has been submitted for examination with my approval as the University advisor.**

**_____**

**MARTHA YIFIRU (PhD)**

**OCTOBER, 2016**

# ACKNOWLEDGMENTS

First of all, I would like to thank my almighty God for supporting and being with me in all of my life. My God, I thank you so much!

My heartfelt thanks should go to my advisor Dr. Martha Yifiru for her constructive comments and guidance. I am thankful to her because her guidance and open comments supported me for the completion of this research.

My special thanks go to Dr. Solomon Teferra, for his sincere clarifications and support which helped me for this study.

I would like to thank the Oromia Radio and Television Organization's employees for their cooperation during audio data collection.

I would like also to thank my wife Alemnesh Temesgen for supporting in all of my study and, my brother Birhanu Gonfa, for facilitating my work. In addition, my thanks go to: Gadisa Hayilu, Firaol Alemayehu, Bekele Abera, Garado Wadajo, and other friends for their support during data preparation and their constructive comments.

Last but not least, I would like to thank the Academia at AAU for their contribution directly and indirectly for success of my study.

**Yadeta Gonfa Gutu**

# TABLE OF CONTENTS

## Contents ……Pages

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ACRONYMS

**AI:** Artificial Intelligence

**ASR:** Automatic Speech Recognition

**BN:** Broadcast News

**CSR:** Continuous Speech Recognition

**CV:** Consonant-Vowel

**CWR**: Connected Word Recognition

**EBC**: Ethiopian Broadcasting Corporation

**ENH**: Ethiopian News Headlines

**FBC:** Fana Broadcasting Corporate

**FFT**: Fast Fourier Transform

**HLM**: HTK language modeling tool

**HMM**: Hidden Markov Model

**HTK:** Hidden Markov Model Toolkit

**IWR:** Isolated Word Recognition

**LPC:** Linear Predictive Coding

**LVCSR:** Large Vocabulary Continuous Speech Recognition

**M.Sc**.: Masters of Science

**NLP:** Natural Language Processing

**ORTO**: Oromiya Radio and Television Organization

**SR:** Speech Recognition

**SRILM:** Stanford Research Institute Language Modelling Toolkit

**TN:** Text News

**TV:** Television

**VOA:** Voice of America

**WER**: Word Error Rate

# ABSTRACT

Speech is a common mode of communication among human beings. Since human beings desired to communicate with machine using speech, developing automatic speech recognition system was the interesting research area. Hence, the main objective of this study was to explore the possibilities of developing a large vocabulary, speaker-independent, continuous speech recognition system for Afaan Oromo using broadcast news speech corpus.

The statistical (stochastic) approach and Hidden Markov Model (HMM) modelling techniques were used in this study. Also tools like HTK, Audacity, and SRILM were also used. The speech corpus collected from different sources like: Oromia Radio and Television Organization (ORTO), Voice of America Afaan Oromo program (VOA), and Fana Broadcasting Corporate (FBC). In general, a broadcast speech corpus consisting of, 2953 utterances (about 6 hours speech) were prepared from 57 speakers (42 males and 15 females). Out of 2953 utterances, 2653 were used for training and the remaining 300 utterances prepared from 12 speakers (9 males and 3 females) which are about 40 minutes long were used for testing the developed speech recognizer. Because of the fact that our speech recognizer system is speaker independent, all speakers who are involved in testing were not involved in training. In addition, a text corpus that is required for language modelling is collected from Bariisaa Afaan Oromo newspaper and bigram language model was developed using the SRILM language modelling tool.

Both context independent (mono-phones based) and context dependent (tri-phones based) acoustic models have been developed. Then, the best performance we have obtained in terms of word error rate was 91.46% WER and 89.84% WER, for context-independent and context-dependent, respectively. Based on the findings and lessons learnt, the researcher concluded that increasing the Gaussian number to 12 and tuning parameters for word insertion penalty 1.0 and grammar scale factors to 15.0 can improve the performance of the system.



**Key words:** Afaan Oromo broadcast news speech recognition, Automatic speech recognition, Afaan Oromo speech recognition, large vocabulary speech recognition

# CHAPTER ONE

# BACKGROUND OF THE STUDY

## 1.1 INTRODUCTION

Speech is a common mode of communication for people throughout their lives. When humans speak, the air passes from lungs through mouth and nasal cavity. This airstream is restricted and changed depending on the position of tongue, teeth and lips. This produces contractions and expansions of the air, an acoustic wave, and sound. Consequently, the sounds so forms are usually called phonemes, and the phonemes are combined together to form words (Rabiner & Juang, 1993).

Human beings desired to communicate with machine such as Computer through their voice with the help of Natural Language Processing (NLP) in general and speech technologies specifically. This is the reason why researchers investigate several ways to enable Computers: record, interpret and understand human speech.

The task of converting from an acoustic wave form to string of word is known as speech recognition. As discussed by Jurafsky and Martin (2007), the process of undertaking an audio wave form as an input and generating a string of words as an output is known as speech recognition (SR) sometimes called Automatic Speech Recognition (ASR). By having the discussion of Jurafsky and Martin (2007), we can say Automatic Speech Recognition (ASR) is the process of when machine took an audio then display the transcribed of uttered words by human. On the other hand, the acoustic wave form is the input whereas the string of word is the output of the ASR system.

Numerous literatures (Jurafsky and Martin 2007; Suman et. al., 2015) told that an Automatic Speech Recognition (ASR) system can be categorized into different types based on several parameters. The most common parameters to categorize speech recognition system are: the nature of utterance, the speaker model, and the vocabulary sizes. Accordingly, based on the nature of utterances the speech recognition system can be classified as isolated speech recognition, connected speech recognition, continuous speech recognition, and spontaneous speech recognition. The second parameter is

considering the speaker model/type. Thus, based on the speaker model the speech recognition system can be classified into two major categories like speaker dependent speech recognition and speaker independent speech recognition. Lastly, according to the size of vocabulary ASR can be classified into three types as small speech recognition, medium speech recognition, large vocabulary speech recognition, and very large vocabulary speech recognition.

## 1.2 STATEMENT OF THE PROBLEM

The objective of speech recognition is to trap human voice in a digital Computer and decode it into corresponding text. The ultimate goal of any automatic speech recognition is to develop a model that converts speech utterance to text or words.

The desire of human beings was to communicate with machine like Computer through their voices using Natural Language Processing (NLP) in general and specifically using speech technology. Therefore, research have been and being conducted in the area of ASR for technologically favored languages. Researchers showed the possibilities of developing Speech Recognition for Broadcast News for European languages (Daniel, 2008) and he conducted his research on Audio Pre-processing and Speech Recognition of Broadcast News for European languages by collecting the corpus from European Portuguese speech and text resources.

In addition, for Ethiopian languages, there are local researchers who started works on speech technologies in general and on speech recognition in particular. For example: Zegaye (2003) developed ASR system for large vocabulary, speaker independent, continuous Amharic speech recognizer using HMM based approach; Solomon (2005) explored various possibilities for developing a Large Vocabulary Speaker Independent Continuous Speech Recognition System for Amharic, and Adugna (2015) developed a spontaneous, speaker independent Amharic speech recognizer by using spontaneous speech such as conversation between two or more speakers; Hafte (2009) tried to design speaker independent continuous Tigrigna speech recognition system and Abdella (2010) tried to explore the possibility of developing prototype speaker dependent speech recognition for Sidaama language using HMM.

Apart from these languages, for Afaan Oromo, local researchers also conducted their studies on isolated word speech recognition system and continuous speech recognition system by Ashenafi (2009) and Kassahun (2010), respectively. Also Teferi (2010) showed the possibilities of developing a Speech Recognition system for Afaan Oromo using Hybrid Hidden Markov Models and Artificial Neural Networks. See chapter 2 section 10 for the detailed review of these works.

Most of the aforementioned researchers have not used broadcast news speech in their studies except Adugna (2015) who developed Amharic ASR using spontaneous speech. To my knowledge, only three researches (Ashenafi, 2009; Kassahun 2010 and Teferi, 2010) have conducted their research in the area of ASR for Afaan Oromo. These works were conducted using medium vocabulary sizes dictionary and read speech corpus. No ASR research is conducted using broadcast news speech corpus for Afaan Oromo. Consequently, Kassahun (2010) forwarded that coming with the increased data size and considering the Afaan Oromo speech from Broadcast News as one source of speech for the development of Afaan Oromo speaker independent continuous speech recognizer is the gap for future work.

For this reason, this study was initiated to investigate on large vocabulary continuous speech recognition by using Afaan Oromo Broadcast News speech.

Therefore, in this research, we investigate the possibility of developing large vocabulary continuous speaker independent speech recognizer system for Afaan Oromo using broadcast news speech.

## 1.3 RESEARCH QUESTIONS

This study tried to answer the following research questions:

a) What are the challenges in developing speech recognition system for Afaan Oromo using broadcast news speech?
b) What are the techniques used to overcome these challenges if any?
c) How do we improve the performance of Afaan Oromo speech recognition system while broadcast news speech corpus was used?

## 1.4 OBJECTIVE OF THE STUDY

The objective of this study can be described as general objective and specific objectives.

### 1.4.1 GENERAL OBJECTIVE

The general objective of the study is to investigate the possibility of developing an Automatic Speech Recognition System for Afaan Oromo using Broadcast News speech.

### 1.4.2 SPECIFIC OBJECTIVES

The specific objectives of the study which help for accomplishment of the aforementioned general objective are listed as follow:

- ✓ To investigate the essential principles of the several approaches, techniques and tools that were used in the study; reviewing several literatures on Automatic Speech Recognition in general, about features of Afaan Oromo, and on how to develop a recognizer system for Afaan Oromo speech.
- ✓ To build acoustic model and language model collecting required audio and text corpus, respectively.
- ✓ Prepare the transcriptions for Afaan Oromo Broadcast News speech corpus.
- ✓ Build a prototype of large vocabulary continuous speech recognizer using HMM for Afaan Oromo Broadcast News
- ✓ Evaluate the performance of the prototype
- ✓ Forward conclusion and recommendations for future work in the area.

## 1.5 SCOPE AND LIMITATIONS OF STUDY

The main aim of this study is to investigate the possibility of developing ASR System for Afaan Oromo using speech corpus from Broadcast News. Accordingly, about 16 hours long speech data was collected from: Oromiya Radio and Television Organization (ORTO) Studio which is located at Adama, Fana Broadcasting Corporate (FBC), and Voice of America Afaan Oromo program (VOA). However, due to time and financial constraints only about 6 hours news have been processed and used for the experiment. The text corpus was collected from ***Bariisaa*** newspaper for language modeling.

The limitation of this study was in both making all transcriptions and text prepared for language modeling was prepared in their lower-case and without any punctuation. Also

the experiment was conducted without identifying the varieties of dialects and age. The varieties of gender was explained only to identify the number of speakers in this study. Another limitation of this study was trigram language model don't used in this study.

## 1.6 METHODS

The research methodology we have used for this particular research was experimental quantitative. Accordingly, under the methodology of this study: reviewing related literatures, data selection and preparation, modeling and evaluations were conducted.

### 1.6.1 REVIEW OF RELATED LITERATURE

To explore the important principles of the several approaches, techniques and tools that were used in the study: reviewing numerous literatures was conducted generally on speech recognition, varieties of speech recognition, using speech corpus from broadcast news. In addition, research conducted on Afaan Oromo speech recognition and other sources have been reviewed.

### 1.6.2 DATA COLLECTION AND SELECTION METHODS

Both text data and audio speech data was collected for this research. The text data was collected from ***Bariisaa*** Afaan Oromo Newspaper. Speech data in audio format was collected from the Oromiya Radio and Television Organization (ORTO), Fana Broadcast Corporate (FBC), and Voice of America Afaan Oromo program (VOA).

We planned to collect audio speech data from Ethiopian Broadcasting Corporation (EBC) Afaan Oromo program but, it is not possible to get audio speech data from EBC for this study. Therefore, generally, we collected about 16 hours of speech audio speech data only from ORTO, FBC, and VOA.

However, by considering the time and budget constraint from speech data collected, the researcher constructed 2953 sentences after audio pre-process was performed and we have prepared the transcriptions for only about 6 hours of 2953 utterances for the experiment of the study. Also after making text normalization we have prepared text corpus of 2,000 sentences for language modeling purpose. The detailed data preparation was described in chapter 4.

The prepared data was required to train the built system and to test the trained system for analyzing the accuracy of that system. Therefore, we have divided the prepared data into two for training and testing purpose. Since large data is required for training; about 2653 utterances were used for training and the remaining 300 utterances were used for testing our speech recognizer system.

### 1.6.3 MODELING TECHNIQUES

As stated by Jurafsky and Martin (2007), Hidden Markov Model (HHM) is more appropriate model for speech recognition because it is very rich in mathematical structure. Therefore, the popular Hidden Markov Model (HMM) was used for modeling in this research.

From several language modelling tools, SRILM was used for the language modelling purpose and the language model built for this study was bigram. Accordingly, we have developed bigram language model by using SRILM for this study.

### 1.6.4 TESTING TECHNIQUES

The trained system should be tested for its accuracy. Then, we have used Word Error Rate (WER) which is a standard evaluation metric for speech recognition system from the HTK toolkit in order to test the accuracy of our developed system.

The word error rate is based on how much the word string returned by the recognizer differs from a correct or reference transcription. Word error could happen in continuous speech recognition system by insertion, deletion, and substitution error words. Low WER represents the high accuracy. Low sample rates and long silence were some factors that increase the WER. The word error rate (WER) can be calculated as:

$$WER = 100X(\frac{S + D + I}{N}) = 100X(\frac{S + D + I}{S + D + H})$$

Where S = the number of substitutions error, D = the number of deletions, I = the number of insertions, H = the number of the correct, and N = the number of words in the reference. However, for reporting the performance of a speech recognition system, we used word accuracy ($WAcc$) instead, which computed from word error rate as:

$$WAcc = 100 - WER$$

## 1.7 TOOLS USED FOR THE STUDY

During the pre-processing of audio speech collected, we have used Audacity 2.1.2 in order to save with .WAV file format the segmented speech and sentence. Audacity cannot segment speech directly. But, we open our audio file with Audacity and segment into sentence. So, speech and sentence segmentation was accomplished manually.

Hidden Markov Models Toolkit (HTK) is a toolkit for building Hidden Markov Models (HMMs). According to Young et.al. (2006), HMMs can be used to model any time series. Also HTK is primarily designed for building HMM based speech processing, in particular speech recognizing, so we prefer to use HTK toolkit in this thesis work. HTK was preferred instead of other toolkits due to the familiarity of the researcher and it is freely available for academic and research use, also it is state of the art in speech recognition task.

Language modelling is important in speech recognition; thus in order to develop bigram language model, we have used the language modelling tool called SRILM. Moreover, tools used in our study were summarized in Appendix A.

## 1.8 SIGNIFICANCE OF THE STUDY

In the advancement of speech technology human beings come to having a desire to communicate with machine such as Computer by using his/her natural languages specifically using voice. The speech recognition has a various advantages as discussed by several researchers. This research is the first of its kind for developing large vocabulary continuous speech recognition system for Afaan Oromo using speech corpus of Broadcast News. Whereas there are some attempt, for isolated Afaan Oromo word recognition and continuous Afaan Oromo speech recognition made earlier by taking speech corpus from read speech.

Since this work is conducted on large vocabulary, continuous, speaker-independent Afaan Oromo speech and the broadcast news used for speech corpus, it shows that the possibilities of developing speech recognizer system for this language by using Broadcast News speech corpus. In addition, using this language in speech technology is one from technologically significant of this research. Since this particular research was conducted on large vocabulary speaker independent continuous speech of Afaan Oromo, a broadcast

news corpus has been prepared. This corpus can be an input to other researchers who will conduct a research in similar area.

Apart from the language (Afaan Oromo), this study was able to show the possibility of developing continuous speech recognition system for Ethiopian languages using broadcast news speech corpus.

## 1.9 ORGANIZATION OF THE STUDY

The whole of this study is organized into six chapters. Chapter one describes about background of the study, statement of the problem, objectives of the study, and significance of this study. In chapter two, we tried to put the reviewed literatures about approaches and tools used in ASR development and to the end of this chapter the related works were presented. In chapter three, we have discussed about Afaan Oromo and alphabets used in the language. Also basic phonetics of the language was discussed in this chapter. In chapter four, we discussed the methodology used for corpus preparation in this study. Chapter five describes about experiment and finding of this study. And lastly in chapter six, we tried to present the conclusion and recommendations forwarded based on the finding of this study.

# CHAPTER TWO

# REVIEWS OF LITERATURES AND RELATED WORKS

## 2.1. INTRODUCTION

With this chapter, we are going to describe the general overview of ASR systems and types of ASR systems. Application area of ASR system, the approaches and tools used in the area of speech recognition were also discussed in this chapter. At the end of this chapter we tried to put the survey of ASR under the related work section.

## 2.2. OVERVIEW OF AUTOMATIC SPEECH RECOGNITION (ASR)

Speech Recognition also known as Automatic Speech Recognition is the process of converting a speech signal to a sequence of words. Automatic Speech Recognition (ASR) works by taking an audio speech as input and then giving the string of words as an output (Young et. al., 2006). For the development of ASR system, the audio format is needed that used as an input parameter, and a large text data was required in order to build language model. The necessary audio could be found from read speech or from real world speech like spontaneous speech and Radio/Television (TV) broadcasts.

Developing ASR from broadcast news speech is not simple task as using a read speech corpus. Because, broadcast news involves non-speech like music background, varieties channels used while recording and the like. Also Daniel (2008) stated that as broadcast news speech includes a speech which is difficult due to their nature of speech.

### 2.2.1. CATEGORIES OF AUTOMATIC SPEECH RECOGNITION (ASR) SYSTEM

Several literatures reveal that as ASR system can be classified to in different categories based on the nature of utterance, speaker type, and vocabulary size (Suman et. al., 2015). Accordingly, let us describe the categories of speech recognition system in the next discussion.

#### Types of Speech Recognition based on nature of utterances

Based on the nature of utterances ASR system can be categorized as: isolated words speech recognition, connected words speech recognition, continuous speech recognition, and spontaneous speech recognition.

**Isolated Words** speech recognition system: Isolated word speech recognition system is a speech recognizer system which recognizes single word. In such systems, the pause or silence is needed before and after the single word. It is suitable for conditions that user is required to give only a single word. Also this type of ASR system is simple and easiest because of the word boundaries are simply identified.

**Connected Words** speech recognition system: This type of speech recognizer system is similar with the recognizer system of isolated words. However, the difference is allowing separate utterances to be run-together by having a minimal pause among words which run together. But it is recalled as pause is used to show the boundary of words that we have discussed for isolated words.

**Continuous Speech** recognition system: This kind of recognizer system allows that users to speak nearly naturally and closest words run together without silence among them. In other words, such speech recognizer system is developed by removing silences among connected words. In continuous speech recognition the word boundary is not simply identified like isolated words speech recognition system. So, this make more difficult to develop speech recognizer from continuous speech (Jurafsky & Martin, 2007).

**Spontaneous Speech** recognition system: The last type of speech recognition system based on utterance is spontaneous speech recognition system which recognizes the natural speech. In this type of speech recognizer system, speech is normally comes suddenly through mouth and mispronunciation may be included. The good example of such type speech may be a conversation between two or more peoples.

**Types of Speech Recognition based on speaker class**

Based on the type of speaker, speech recognition system can be categorized as speaker dependent and speaker independent.

**Speaker dependent** speech recognition system: this type of speech recognizer system is developed for particular speaker. Speaker dependent speech recognition system is more accurate since developed for individual speaker. Also, typically it is easier to develop and cheaper when compared with speaker independent. However, such type of system is not

flexible (elastic) as speaker independent speech recognizer system. Because, it depends on the speaker's characteristics like accent, dialect and the other (Suman et. al., 2015).

**Speaker independent** speech recognition system: the other type of speech recognizer system based on speaker class is speaker independent speech recognition system. This type of speech recognition system was not enforced the individual speaker like speaker dependent speech recognition system. Speaker independent speech recognition system can recognize a variation of speaker. However, implementation of such system is not easy and cheap like speaker dependent. Similarly the accuracy of this system is lower than the speaker dependent.

**Types of Speech Recognition based on vocabulary size**

The last parameter to categorize speech recognition system is vocabulary size. Based on the size of vocabulary speech recognition system can be classified as small vocabulary, medium vocabulary, large vocabulary, and very large vocabulary which we will discussed as following:

**Small Vocabulary** speech recognition system**:** This type of speech recognition system contains 1 to 100 words and mostly suitable for developing command-control.

**Medium Vocabulary** speech recognition system: A medium vocabulary speech recognition system contains 101 to 1000 words.

**Large Vocabulary** speech recognition system: A speech recognition system developed from 1001 to 10,000 words is known as a large vocabulary speech recognition system.

**Very large vocabulary** speech recognition system: Such type of speech recognition system contains more than 10,000 words.

## 2.3. HISTORY OF AUTOMATIC SPEECH RECOGNITION (ASR)

Generally, the development of ASR system started firstly by the efforts of the 19th century. As mentioned by Rabiner and Juang (2004), the speech recognition was started by building a system for isolated word recognition system for a single speaker in Bell Laboratories in the year of 1952. This was done by designing to develop the ASR system for the ten digits (one to nine and zero).

The developed speech recognition system in 1960's was for small vocabulary (10 – 100) by using acoustic-phonetic approach. The next evolution ASR system was by adding the size of vocabulary or shifting from small to medium (101 – 1000) and applying template based/pattern recognition approach in 1970's. This system is also developed for connected digits and continuous speech in addition to the isolated words recognition system (Rabiner & Juang, 2004).

The other evolution of ASR system is coming with the idea of large vocabulary which was started in 1980's. The system was developed for connected word and continuous speech by applying statistical approach. Then, the continuity of large vocabulary; also for very large vocabulary ASR system was investigated in 2000's for spoken dialogs using multimodal dialogs approach (Rabiner, 1989).

## 2.4. APPROACHES OF SPEECH RECOGNITION

The selected approaches of speech recognition can be varied based on the varieties of speech recognition and the desired type of speech recognition system. This is the reason why different researcher used different approaches in their work. As mentioned by Rabiner and Juang (1993), there are numerous of approaches which can be undertaken in the work of ASR. Accordingly, most of the time there are three basic speech recognition approaches namely:

a. Rule Based (Acoustic-phonetic) approach
b. Statistical Based (Stochastic) approach
c. Artificial Intelligence approach

### a. Rule-Based (Acoustic-phonetic) approach

One approach used in the work of ASR development is Acoustic-phonetic also called rule-based approach. This approach used the knowledge of phonetics and linguistics to guide search process. Usually, some rules are defined by expressing that might support to decode: Phonetics, phonology, Syntax and Pragmatics.

In this approach, the speech recognition systems are based on finding speech sounds and providing appropriate labels to sounds. This is the basis of the acoustic phonetic approach which assumes that there exist finite, distinctive phonetic units (phonemes) in

spoken language and that these units are broadly characterized by a set of acoustics properties that are manifested in the speech signal over time. However, this approach has not been widely used in the most of ASR works. As mentioned by Rabiner and Juang (1993), due to the following drawbacks it can perform poor.

- the difficulty of expressing rules
- the difficulty of making rules interact
- the difficulty of knowing how to improve the system

Because of the above limitations, the AP approach has not achieved the same success in practical systems as have alternative method. But its underlying ideas are still used in the artificial intelligence based recognizers.

### b. Statistical- Based (Stochastic) Approach

The other approach is statistical approach that are most commonly applied in modern speech recognizers due to its strength (Rabiner & Juang, 1993). This approach is also based on machine learning and HMM model. In this approach, variations in speech are modeled statistically. In addition, this approach represents the current state of the art and used for general purpose speech recognition systems which are based on statistical acoustic and language models.

However, there are the factors that distinguish different pattern recognition approaches like: the types of features, the choice of templates or models for reference patterns, and the method used to create reference patterns and classify unknown test patterns.

### c. Artificial Intelligence Approach

The Artificial Intelligence (AI) approach is a hybrid of the acoustic-phonetic and statistical approaches in speech recognition methods. This approach tries to mimic the human intelligence in visualizing, analyzing and decision making progress on the measured acoustic features. As discussed by Rabiner and Juang (1993), the main idea of AI is to collect and employ knowledge from a number of sources for solving the problem in question. The knowledge sources (KS) are wide ranging from fields of acoustic, lexical, syntactic, semantic and pragmatic knowledge.

The most important techniques undertaken in this approach is the use of an expert system for segmentation and labeling of the acoustic signal, learning, and adaptation over time, the use of artificial neural network (ANNs) for distinction between similar sound classes and learning the relations between all known inputs and phonetic data. Also the neural network could represent a separate structural approach to speech recognition orregardedasanimplementationarchitecturepossiblyincorporatedinanyofthethree classical speech recognition approaches (Rabiner & Juang, 1993). However, the modern methods are mostly applying hybrids of ANNs and HMMs.

## 2.5. GENERAL OVERVIEW OF MARKOV MODEL

A Markov chain which sometime called observed Markov Model is a special case of a weighted automaton in which weights are probabilities(the probabilities on all arcs leaving a node must sum to 1) and in which the input sequence uniquely determines which states the automaton will go through (Jurafsky & Martin, 2014). A Markov chain is only useful for assigning probabilities to unambiguous sequences; because it can't represent inherently ambiguous problems.

On the other hand, a Markov chain is a model which is useful when we need to compute a probability for a sequence of events that we can observe in the world. But, in many cases the events we are interested may not be directly observable in the world. This is a reason that we need a Hidden Markov Model. Because, a hidden Markov model (HMM) allows us to talk about both observed events that we see in the input and hidden events that we think of as causal factors in our probabilistic model.

### 2.5.1. DEFINITION OF HIDDEN MARKOV MODEL (HMM)

A Hidden Markov Model (HMM) is especially embedded stochastic process with underlying stochastic process which is not directly observable, but can be observed only through another set of stochastic processes that produce the sequence of observations (Rabiner & Juang, 1993). Also an HMM is very rich in mathematical structure.

### 2.5.2. ELEMENTS OF HMM

HMM involves several elements which generate the observation sequences. And an HMM is characterized by these elements. As stated by Rabiner (1989), these elements of an HMM are discussed as following:

i. Hidden states in the model $N$: the states are hidden, for many practical applications. There is some physical significance attached to the states or to the set of the states of the model. The individual state donated by: $S = \{S1, S2, .., SN\}$, and the state at a time $t$ as $qt$.
ii. Distinct observation symbol per state $M$: the observation symbols correspond to the physical output of the system when modelled. The individual symbols are denoted as: $V = \{v1, v2, \dots, vM\}$.
iii. State transition probability $\boldsymbol{A}$: $aij = P[qt + 1 = Si | qt = Sj], 1 \leq i, j \leq N$
iv. Observation symbol probability distribution in state $j, Bj(K) = P[Vk \; at \; t | qt = Sj]$
v. The initial state distribution $\pi = \pi i$, where $\pi i = P[q1 = Si] \; 1 \leq i \leq N$[2]

Given appropriate value of $N, M, A, B$ and π, HMM can be used as generator to give an observation sequence as: $O = O_1 \; O_2 \; O_3 \dots O_T$.

In order to model the changing statistical characteristics that are only probabilistically established through actual observations, HMMs use a process. The sequence of state is hidden and only observed through another set of observable stochastic processes. Each hidden state of the model (or the transition between states) is associated with a set of output probability distribution or continuous Probability Density Functions (PDF).

Also a HMM consists of a number of states. Each state $j$ has an associated $o$ observation distribution $bj(ot)$ which determines the probability of generating observation $ot$ at time pair of states $i$ and $j$ has an associated transition probability $aij$. In HTK the entry $\boldsymbol{s}$ the exit state $N$ of $N$ state HMM are non-emitting.

Figure 2.1 shows the hypothetical example for five states that HMM undergoes while a change of states according to a set of probabilities associated with the states.

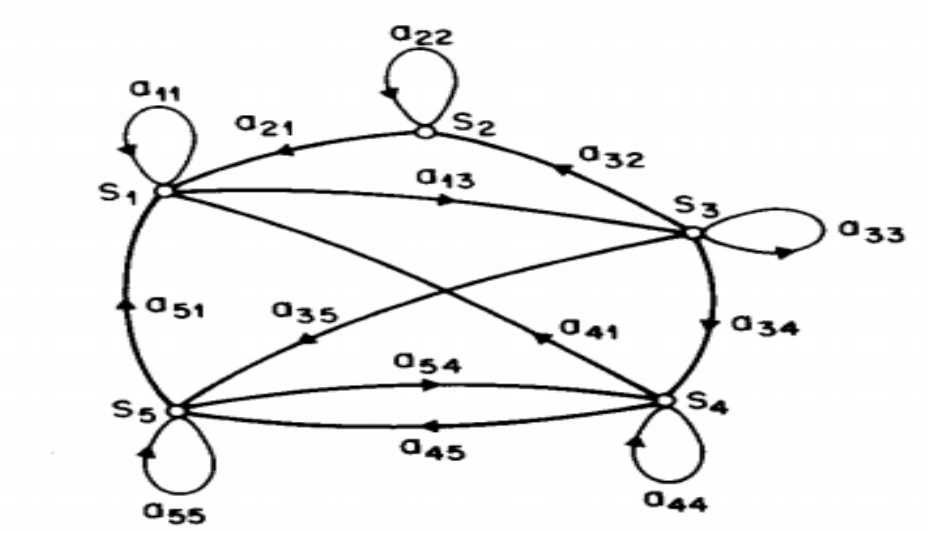


Figure 2.1 Markov Chains with 5 states (taken from Rabiner, 1989)

### 2.5.3. BASIC ASSUMPTIONS OF HMM

Literatures revealed that there are three basic assumptions while using HMM that one needs to consider (Rabiner, 1989); (Rabiner & Juang 1993). These assumptions are made for mathematical and computational traceability. The three assumptions of HMM are: Markov Assumption, Stationary Assumption, and output independence assumption that we are going to describe as follows.

#### a. Markov Assumption

The first assumption is Markov Assumption which states that history has no influence on the chain's future evolution if the present is specified. In other words, it is assumed that the next state is dependent only upon the current state.

This assumption is called the Markov assumption and the resulting model becomes actually a first order HMM. Even though the first order HMMs are the most common, some attempts have been made to use the higher order HMMs for speech recognition and other related stochastic process modelling.

In a first order of Markov assumption, the probability of an observation at time $n$ only depends on the observation at timen-1; and in second order of Markov assumption, the probability of an observation would have the observation at time n depend on n -1and n-2. So, a higher order Markov assumption allows more contexts into the grammars.

### b. Stationary Assumption

The second basic assumption of HMM is the stationary assumption which states that, the state transition probabilities are independent of the actual time at which the transitions take place. Therefore, this assumption approves that the transitions are independent of the point of time at which the transition took place.

### c. Output Independence Assumption

The third basic assumption of HMM is output independence or sometimes called as output independence hypothesis. This assumption states that neither chain evolution nor past observations influence the present observation if the last chain transition is specified. This means, the current observations or outputs are statistically independent of the previous observations or outputs.

Unlike the other two assumptions of HMMs, the output independence assumption has limited validity and sometimes considered as a sever weakness of HMM. In the next section the three problems of HMMs and their respective solutions are elaborated.

## 2.5.4. THREE BASIC PROBLEMS FOR HMMS

### i. Problem 1: Evaluation Problem

Problem 1 is evaluation problem i.e. given an observation sequence and a model, how do we compute probability that the observed sequence was produce by the model. In other words, an observation sequence $O = O_1\ O_2\ \dots\ O_T$, and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of observation sequence given the model. We can consider it as scoring problem. If we have to choose between many competing models then one with maximum probability will give better result.

### ii. Problem 2: Hidden State Determination (Decoding)

Problem 2 is one which attempts to uncover the hidden part of the problem i.e. given an observation sequence and a model, how do we choose corresponding state sequence which is optimal in some meaningful sense. In other words, an observation sequence $O = O_1\ O_2\ \dots\ O_T$, and a model $\lambda = (A, B, \pi)$, $Q = Q_1\ Q_2\ \dots\ Q_T$, shows that how do we choose the

matching state sequence. There is no correct solution to this problem, but in practice we usually use optimal criterion to find best possible solution.

### iii. Problem 3: Learning

Problem 3 is learning problem i.e. one in which we try to optimize model parameter so as to best describe as to how given observation sequence comes out. In other speaking, how do we adjust the model parameter $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$. The observation sequence used here are called "training" sequence since it is used for training HMM. Training is one of the crucial elements of HMM. It allows us to adjust model parameter as to create best model for given training sequence.

## 2.5.5. SOLUTIONSOFTHE THREE PROBLEMS

When the problem is given, solving them using efficient algorithm is important. Therefore, the following description shows the solutions to the three problems of HMMs.

### i. Forward Algorithm for Evaluation Problem $[\boldsymbol{P(O|)}]$

We want to find $P(O|\lambda)$, given an observation sequence $O = O_1, O_2, \ldots, O_T$ and a model. The most straight forward way to find the solution is enumerating every possible state sequence of length $T$. Consider one such state sequence $Q = Q_1, Q_2, \ldots, Q_T$ such that $Q_1$ produces $O_1$ with some probability, $Q_2$ produces $O_2$ with some probability and so on. While using chain rule and when the order of chain rule is $NT$, at every $t = 1, 2, \cdots, T$, there are $N$ possible states which can be reached.

$$P(O|\lambda) = \sum_{q_1, q_2, \cdots, q_T} \pi_{q_1} b_{q1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T)$$

### ii. Viterbi Algorithm for Decoding Hidden State Sequence $\boldsymbol{P(Q, O|\lambda)}$

There are various ways in which optimum solution can be calculated. The difficulty lies in the definition of optimum state sequence. One way to find optimum state sequence is to choose the states $qt$ which are individually most likely. But, this way has serious flaw in the sense that if two states $i$ and $j$ are selected such that $aij = 0$, then despite of the most likely state at time $t$ and $t + 1$, which is not valid state sequence.

So deciding optimum criteria is very crucial. A way to find the single best state sequence is using dynamic programming algorithm, which is called Viterbi algorithm.

Viterbi Algorithm has widespread applications in speech recognition and also used in most communication devices (e.g. cell phones) to decode messages in noisy channels. Decoding of the speech (the term for what happens when the system is presented with a new utterance and must compute the most likely source sentence) would probably use the Viterbi Algorithm to find the best path.

### iii. Baum-Welch Algorithm for Learning

Baum-Welch is an iterative procedure and each-iterations of Baum-Welch are guaranteed to increase the log-likelihood of the data. Baum-Welch also works by maximizing a proxy to the log likelihood, and updating the current model to be closer to the optimal model. But of course, convergence to the optimal solution is not guaranteed.

Most challenging of all is to adjust the model parameter $(A, B, \lambda)$ to maximize the probability of the observation sequence given the model. Given any finite observation sequence as training data, there is no optimal way of estimating the model parameters. But there are some heuristic procedures which try to find local optimization over global optimization.

## 2.6. AUTOMATIC SPEECH RECOGNITION PROCESS

As we have discussed before, automatic speech recognition (ASR) is the process of converting an acoustic signal containing human speech into a words transcript. Figure 2.2 shows the basic components of a typical ASR system. The digitized speech signal is first transformed into a set of features at a fixed rate (typically once every 10-20 msec). The features, represented as a sequence of vectors are passed to a decoder that uses them to search for the most likely word candidates, making use of constraints imposed by the acoustic, lexical and language models.

Acoustic models also called observation probability that consist of statistical representations of distinct sounds that compound the words of a language. The lexical model refers to a pronunciation dictionary where each word entry has an associated

phonetic transcription. Language model is also a-priori probability that contain a large list of words associated with their occurrence probability in a given sentence.

As we have seen from Figure 2.2, the speech signal is the input to speech analysis for extracting feature which can be taken to the decoder. Also the lexicon, acoustic model, and language model were also given to the decoder; then the transcribed text was the final output of the system. These components were also described in next section.

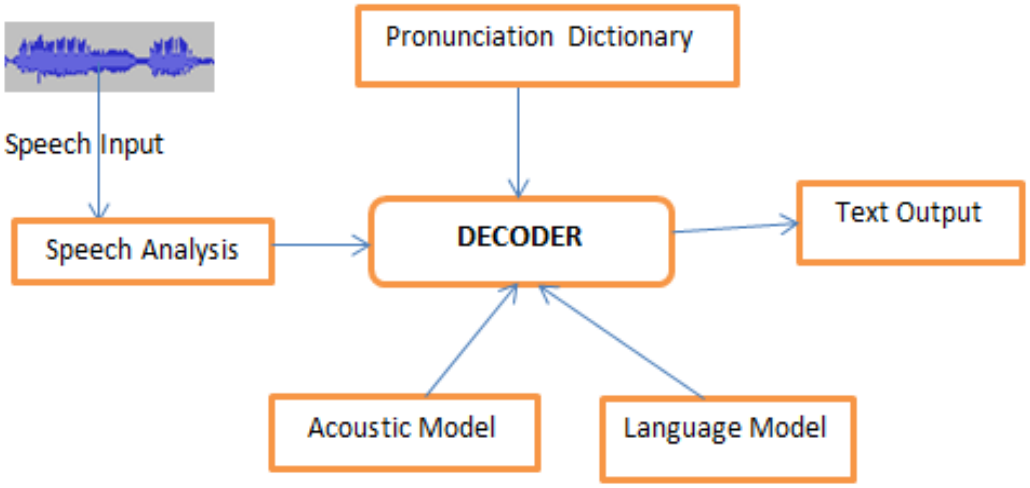


Figure 2.2 Automatic Speech Recognition Components in Process

### i. Language Model (Grammar)

Several literatures reveal that as a language model is used in speech recognition systems and automatic translation systems to improve the performance of such systems. The language model for speech recognition is one of the important aspects that we have to consider here. Therefore, one the dominant language model used in line with the HMM approach is the N-gram language model. Typically language models contain a very large list of words and their probability of occurrence in a given sequence, a large textual data was required for language modelling purpose.

For general purpose speech recognition, the language model can be an N-gram model of text learned from a corpus of written sentences. But, spoken language has different characteristics than written language. So, it is better to get a corpus of transcripts of spoken language. For task specific speech recognition, the corpus should be task specific. For that reason, Afaan Oromo newspaper ***Bariisaa*** was selected with this study. In

addition to this, the textual news collected from the sources that have the similarity with source of the audio was collected.

Grammars are a much smaller file containing sets of predefined combinations of words. So each word in a Language Model or Grammar has an associated list of phonemes (which correspond to the distinct sounds that make up a word). In addition, $N$ −gram model is defined as a Markov chain of order $n - 1$.

### ii. Acoustic Model

Acoustic model is a file that contains a statistical representation of each distinct sound that makes up a spoken word. This contains the sounds of each word used in our grammar for this work. In order to determine what sounds will be produced after giving string of words which is uttered Acoustic Modelling should be undertaken. Accordingly, for all possible combinations of word strings $W$ and observation sequences $O$ the probability $P(O|W)$ could obtainable sequences $O$ the probability $P(O|W)$ must be obtainable (Alexander, 2003). So, the words in our grammar were giving the sequence of sounds it must listen for. Accordingly, from the listened sequence of sounds that make up a particular word, the textual representation of the word was returned. Thus, when listening for sounds of words, it is actually listening for the sequence of sounds that make up one of the words we defined in our Grammar.

In general speaking, the Grammar and the Acoustic Model work together. Because of this, most common words that might be used in a Grammar are already included in their Acoustic Model. Therefore, when we train our Acoustic Model to recognize the word/phrase/sentence in our work it is actually listening for the phoneme sequences.

The other point what we should have to take into account is when creating our own Acoustic Models and Grammars, we need to make sure that all the phonemes that make up the words in our Grammar are included in the Acoustic Model.

### iii. Decoder

Decoder is a software program that takes the sounds spoken by a user and searches the Acoustic Model for the equivalent sounds. When a match is made, the Decoder determines the phoneme corresponding to the sound. It keeps track of the matching phonemes until

it reaches a pause in the user speech. It then searches the Language Model or Grammar file for the equivalent series of phonemes. If a match is made it returns the text of the corresponding word or phrase to the calling program.

## 2.7. AUDIO PRE-PROCESSING

As it is discussed in HTK Book Young et. al. (2006), the collected read data should be converted into the correct required format using tools which HTK provides. Furthermore, Daniel (2008) discussed that the processing activities that can be carried out meanwhile audio pre-processing are:

i. Audio or speech segmentation,
ii. Speech or non-speech discrimination,
iii. Gender classification, and
iv. Sentence segmentation

### i. Audio or Speech Segmentation

The process of identifying the boundaries among words, syllables, or phonemes in spoken natural languages is known as speech segmentation. The boundaries of words and sentences are based on the pause which occurs before and after each that sentences. Daniel (2008) suggests that as audio and speech segmentation will constantly be needed to break the continuous audio stream into manageable chunks. The main goal for audio segmentation is to split the input audio stream into acoustically similar segments. Therefore, we have done the speech segmentation using Audacity.

### ii. Speech or Non-Speech discrimination

After the process of audio segmentation the segmented speech should be classified into speech or non-speech using a speech or non-speech discriminator. Daniel (2008) stated that as identifying speech and non-speech is very important for the rest of the processing activity because wasting time trying to recognize audio segments that do not contain valuable speech such as too much noise and pure music which are non-speech is not interesting. In this study, after performing the speech or non-speech discrimination, non-speech was discarded because of it was not required for our work.

### iii. Gender Classification

A gender classification is used as a mean to improve speaker clustering based on the maximum likelihood whether the anchor is male or female. Daniel (2008) tried to define the concept of gender classification as, "gender detection distinguishes between male and female speakers". Accordingly the task of gender classification is performed manually to classify male speakers and females separately.

### iv. Sentence segmentation

From audio data simply identifying the boundary of sentence is not simple like identifying the sentence boundary from text data. However, it was discussed in Daniel (2008) as it is a simple guess that assumes a speech pause will correspond to an end-of sentence. But in broadcast news the news reporters (anchors) not always do a breath pause at the end-of sentence points. So, this should be done carefully and referring the text. Because the major source for incorrect sentence boundaries came from incorrect sentence segmentation.

There are various ways of sentence segmentation: manual and automatic. But we couldn't get the appropriate tool which segment Afaan Oromo sentences from audio data in our work. So, segment sentences manually.

## 2.8. APPLICATION AREA OF ASR SYSTEMS

Like other systems, ASR systems are useful in various aspects of life. So ASR systems can provide a numerous applications such as in telephony system, in command control system, and in dictation system. Some specific examples of application of ASR may include but not limited to the following:

- Dictation systems - for formal document preparation in legal or medical services.
- Interactive voice response - for callers who do not have tone pads, for the automation of call centres, and for access to information services such as stock market quotes.
- Telecom assistants - for repertory dialling and personal management systems.
- Process and factory management - for stocktaking, measurement and quality control.
- Command and Control systems - for accessing computers with the help commands.

- As an assistive technology for disabled people -people with disabilities can benefit from speech recognition programs.
- Medical Documentation-in health care sector, speech recognition can be implemented in front-end or back-end of the medical documentation process.

## 2.9. AUTOMATIC SPEECH RECOGNITION TOOLS

According to different literature, there are several tools used in the development of Automatic Speech Recognition (ASR). The tools required for the development of automatic speech recognition (ASR) systems are categorized into two based on their accessibility. Firstly there are commercial systems which provide ASR tools such as AT&T Watson, Microsoft Speech Server, Google Speech API, and Nuance Recognize. On the other hand, the patented systems offer little control over the recognizer's features, and limited built-in integer-ability into other software, and leading to a releasing of a great number of open-source automatic speech recognition (ASR) systems. Like SPHINX, KALDI, and HTK. Since this research is for academic purpose, the open source ASR tools are preferred in this work. These open-source ASR tools are available free of charge for research and education.

### i. SPHINX

The SPHINX system has been developed at Carnegie Mellon University and presently there are a several versions of it such as SPHINX 2, 3, 3.5 and 4 decoder and Sphinx Train which used for training purposes (Juraj, 2004). Also SPHINX has an acoustic model and language model with a priori probabilities for transition between phones and words, respectively. However it is not well documented features, so the SPHINX documentation is poor when relatively compared with HTK. In addition, the SPHINX system has limitation in model structure in the training process. The main drawback is only one structure which allowed for all models.

### ii. KALDI

The work on KALDI started during the 2009 Johns Hopkins University summer workshop project titled "Low Development Cost, High Quality Speech Recognition for New Languages and Domains," where we were working on acoustic modeling using subspace Gaussian mixture model (Povey et. al., 2012). KALDI is an open-source toolkit for speech

recognition written in C++ and licensed under the Apache License v2.0. This system is specifically designed for speech recognition research application.

As discussed by Povey et. al. (2012), the basic advantage of KALDI is such modular source, open license, some of example scripts, and optimized for LVCSR tasks. However, its limitations are a little hard to work with on windows, user defined is not allowed because of its commands and defaults, and knowledge on shell scripting was required. The other main drawback of KALDI is decoding with a complex cyclic grammar is not easy because of missing decoders or it use the temporary decoder in KALDI and KALDI decoders require specific properties of the grammar it can decode. The KALDI decoders do not know about the HMM topology or GMMs, they only know about a decode-able interface. Therefore, it needs still to incorporate it with other kind of acoustic models to make it easier.

### iii. Hidden Markov Toolkits (HTK)

The Hidden Markov Toolkit (HTK), which is developed by the Speech Group at the Engineering Department Cambridge University, is the most widely used tool for speech recognition (Young et. al., 2006). HTK is an open-source research toolkit available for free of charge from link: http://htk.eng.cam.ac.uk/. The main advantages of the HTK over the other tools are: it is a complex system that covers all development phases of a recognition system, system is regularly updated to catch up with the latest advances in the field of recognition, and it is well documented both theoretically and practically when it is compared with SPHINX and KALDI.

HTK also include several tools which are implemented to perform tasks like manipulation of transcriptions, coding data, and evaluation of the result. In addition, HTK contains a set of library modules and tools available in C source form and these tools provide sophisticated facilities for the works in developing ASR systems. Therefore, HTK was preferred instead of other toolkits due to the knowledge of the researcher, also it is state of the art in speech recognition task.

## 2.10. RELATED WORKS

Automatic Speech Recognition (ASR) has been the focus of many researchers for numerous decades. Also ASR becomes an interesting research area in the world and it

requires still more investigation because of the variety of languages spoken in the world. In this section, reviews of related international as well as local works are presented.

Murat (2003) conducted a research on large vocabulary continuous speech recognition for Turkish using HTK. In his work, he used the un-segmented data; that consisted of 7650 utterances spoken by 104 male and 89 female speakers for training. The researcher has performed five experiments like the IWR (Isolated Word Recognition) task, a CWR (Connected Word Recognition) system with no grammar, a CWR system with a simple grammar that is a CSR (Continuous Speech Recognition) system, and the fifth experiment was performed in order to test using bigram language model which developed by using HTK language modeling tool (HLM).

In his thesis, five experiments were done. The first one was the IWR (Isolated Word Recognition) task. In the first experiment to perform an IWR task, he cut 40 speech segments from test data, which contained only one word. In the second one, he tested a CWR (Connected Word Recognition) system with no grammar. This means, every word can follow another one employing no rule in the second experiment. In the third experiment, he tested a CWR system with a simple grammar that he designed. Accordingly, follower words are determined according to the grammar he designed. The fourth experiment was related to a CSR (Continuous Speech Recognition) system, in which the cross-word expanded network is based on bigrams that include stems and endings. The fifth experiment was performed in order to test the bigram language model that is actually proposed in his thesis.

The test utterances contain 220 sentences (have a vocabulary size 1168 words) that are randomly selected from the text corpus. In the experiment these sentences were continuously spoken by 6 speakers (4 male, 2 female). After performing the five experiments by comparing the results of correct sentence recognition rate (CSRR) 20.09% and 30.90% given in experiment 4 and experiment 5, respectively. This result shows that experiment 4 performed better than experiment 5 since the correct sentence recognition rate is bigger for experiment 5 than experiment 4. To the end since he applied no smoothing algorithm to the language model built; he has concluded as applying a smoothing technique would be better.

Daniel (2008) conducted a research on Audio Pre-processing and Speech Recognition for Broadcast News. The researcher has collected the speech and text corpus from ALERT-SR and WEBNEWS-PT, respectively for European Portuguese. These corpora are founded mainly by news reports from two sources: television Broadcast News (BN) and web Text News (TN). The approach used and language built in the study was Artificial Neural Networks and 4-gram language model, respectively. The BN corpora collected were mostly composed by news and shows which transmitted by the Portuguese public television broadcasting company and the TN corpus collected from several European Portuguese daily newspapers and web journal articles which is called WEBNEWS-PT and the researcher used the ALERT-SR system which has less than 50 hours of speech for training.

The researcher achieved at the result which contains a higher Word Error Rate (22.1%WER) than the state of the art ASR systems for English and French. In addition, building a speaker adapted acoustic models was a more effective approach to reduce WER as the researcher suggestion. Because like news anchors which have a lot of speech could be captured from built speaker adapt acoustic model. These is therefore it was suggested that state of the art ASR systems for English and French were trained at least with 190 h of manually labeled training data. At the end the researcher recommend that a possible future direction for improving ASR performance is to develop a better feature extraction module combining the outputs of different features.

There are also a number of local researchers who conducted research on speech recognition for Ethiopian languages. Certain of them which we have undertaken considering the approach and tools they have used in their work were going to describe.

Zegaye (2003) developed ASR system for large vocabulary speaker independent continuous Amharic speech recognizer using HMM based approach and HTK tools. He used a corpus database comprised of 8000 utterances that were developed by Solomon Tefera. The corpus he used for training contained 8000 sentences spoken by 80 people, each having read 100 utterances which involved of all Amharic phonemes in many phonetic contexts. And the ratio between male and female speakers is almost fifty-fifty. The language model he constructed was a bigram language model.

For development testing and evaluation purpose 36 sentences (i.e., 18 sentences read by different 15 persons for both testing and evaluation) were used. In other hand 270 utterances used for development of testing and 270 utterances used for evaluation purpose and these 540 utterances are added by the researcher. At the end of his experiment, he achieved the accuracy of 79% word level correctness, 76.18% word accuracy, and 30.01% sentence level correctness. Finally, he concluded that as the result of his experiment is a proof of the fact that it is possible to construct an Amharic large vocabulary, speaker independent continuous speech recognizer using the HTK toolkit and the HMM modelling technique.

Solomon (2005) tried to explore the various possibilities for developing a Large Vocabulary Speaker Independent Continuous Speech Recognition System for Amharic by developing an Amharic speech corpus that can be used for several kinds of investigations into the nature of spoken Amharic. By using the speech corpus, he developed ASR system for Amharic based on Hidden Markov Models (HMM). The researcher also used a Pattern Recognition approach with the most successful and popular method HMM.

The researcher acquired 100,000 sentences from EthioZena which is before called as Ethiopian News Headlines (ENH). After making some corrections on spelling and grammars; avoiding a sentences which have more than 20 words that create difficulties for the readers, about 72,000 sentences were selected. On the other hand, these selected sentences have been manually checked for grammaticality, spelling, foreign words, abbreviations, and others corrected manually. From the text database, 53 sentences were selected to create phonetically rich and balanced text database, for speaker adaption set. The remaining 10,000 sentences were selected to create phonetically rich and balanced text database for training set. The text corpus has been read by 124 native Amharic speakers to prepare a read-speech corpus for the study. This means, the recorded training set consists of a total 10850 different sentences. These training sets were read by 80 speakers of Addis Ababa dialect; test and speaker adaptation sets were read by 20 other speakers of Addis Ababa dialect and 4 speakers of the other dialects. Also the researcher have trained bigram language models using the HTK statistical language model development modules. From the tri-phone models tested, the best result obtained has a word recognition accuracy of 91.31% at a speed of 3.8 minutes per sentence. Thus, the researcher concluded that for Amharic modelling CV syllables, as represented by the

orthographic symbols, is the better alternative to prevailing modelling unit of elementary sounds, like phones.

Adugna (2015) tried to develop a spontaneous, speaker independent Amharic speech recognizer by using speeches such as conversation between two or more speakers in the form of conversation. For his work, the speech data with 44100 Hz sampling rate obtained from web that are recorded in local Medias. The speech data used for both training and testing are conversational speeches which are made between two or more speakers. The popular modeling techniques Hidden Markov Model (HMM) was used and HTK toolkit was also employed in the work of (Adugna, 2015). For his work he has collected the speech data from web and the collected data are transcribed manually. The language model used in his work was bigram language model. From the collected data, 2007 sentences which are uttered by 36 peoples who are different in their age group and sex are used for training. So, this training data consists of 9460 unique words and the duration of speech is around 3 hours and 10 minutes. He also used 820 unique words from 104 utterances (sentences) uttered by 14 speakers (5 females and 9 males) for testing. However, 10 test sentences speakers were involved training, and 4 speakers were not involved. The best performance result that he has got was 41.60% and 23.25% of word accuracy for test data from speakers those are involved in training and speakers those do not involved in training, respectively. In addition, 39.86% of word accuracy for test data from both speakers those are involved in training and do not involved in training was obtained.

Hafte (2009), tried to design speaker independent continuous Tigrigna recognition system. The researcher used the HMM as modeling techniques and the work is done by HTK toolkit. The text data was collected from sentences which are selected from one fiction book namely “Hezikenabey” and one newspaper namely “Weyn”. As indicated by the researcher, phonetically rich and balanced collections of sentences were selected from the sources. In the process of text selections some corrections such as spelling and grammar errors has been corrected, long sentences are shortened and numbers have been textually transcribed have done manually. After these corrections, 300 sentences were made ready for reading. The utterances of 12 persons who are in the age range of 20 – 60 provided 300 utterances (i.e., each person read 25 sentences). The speech data is recorded at a sampling rate of 16 KHz and the recorded speech is then converted into Mel

Frequency Cepstral Coefficient (MFCC) vectors. The researcher conducted the experiment by splitting datasets into two for training and testing purpose. Accordingly when 83.33 % of the prepared data was used for training purpose and the remaining 16.67% of the data was used for testing. So the best performance result obtained are 60.32%,58.38%, and 20 % for word level correctness, word accuracy, and sentence level correctness, respectively.

Abdella (2010) tried to explore the possibility of developing prototype speaker dependent speech recognition for Sidaama language using Hidden Markov Model (HMM). The researcher used HMM modeling technique for modeling and Java based speech recognizer tool called the Sphinx Systems to build the acoustic models and for testing the recognition performance. By consulting linguistics experts, 450 representative words are selected and each of the words was recorded from a single speaker because the planned work is for speaker dependent. Out of 450 words, 66.67% of the recorded words were used for training the acoustic models whereas the remaining 33.33% of words were used for testing the performances of the constructed acoustic models. In addition, from 300 or (66.67%) words used for training the HMM acoustic model 100 words selected arbitrary to test the constructed models. After testing the constructed models the researcher achieved hopeful results. Accordingly, for context dependent tri-phone based model the achieved recognition accuracy of 73% and 68% for test words selected from training set and accuracy for test words not included in training set, respectively. But for context independent mono-phone based model the achieved recognition accuracy of 65% and 48% for test words selected from the training set and accuracy for tests words not included in training set, respectively. By comparing the results obtained the researcher concluded that the tri-phone based model is better than the mono-phone based context independent model for the language.

In addition to the above discussed speech recognition was conducted particularly for Afaan Oromo. Ashenafi (2009) conducted a study on Speech Recognition System for Afaan Oromo. The researcher focused on isolated words and also used the HMM model and an open source speech recognition toolkit Sphinx4. A researcher prepared 50 Afaan Oromo words as corpus by consulting the domain experts. Then these words were read by 20 persons and 1000 utterances of Afaan Oromo isolated words were obtained. In his study, when the 66.67% of the data was used for training, the remaining (33.33% of the

data) was used for testing purpose. At the end, the word level accuracy achieved by the researcher's work was 82.83% and 81.081% for context dependent phoneme based model and context independent word based, respectively.

Kassahun (2010) tried to develop a Continuous, Speaker Independent Speech Recognizer for Afaan Oromo. The main purpose of his work was to explore the possibility of developing continuous Afaan Oromo speech recognition system by the use of the already available tools and techniques. The researcher used HMM model and sphinx system (Sphinx train for training and Sphinx4 for decoding). He selected 70 Afaan Oromo long words, phrases, and simple sentences that were read by 30 people who are different by their age and gender. Thus, generally he prepared a corpus consisting of 2100 utterances. Like Ashenafi (2009), he also used the 66.67% of the data for training; and the remaining (33.33% of the data) is used for testing purpose. The performance level which achieved by the researcher was 68.514% with sentence accuracy of 28% and 89.459% with the sentence accuracy of 42% for word level and tri-phone based recognition system, respectively.

Teferi (2010) tried to explore a speech recognition for Afan Oromo and the possibility of its applicability. The researcher used a hybrid HMM/ANN model to see its effectiveness compared to the more common HMM. For developing and testing the hybrid ASR system, CSLU hybrid toolkit was used along with other tools used for recording and labeling the speech corpus. The experiment was conducted on recognition of limited vocabulary. A total of hundred Afan Oromo word are selected and all the 29 phonemes of the language are considered during the selection. To make the recording easy, the words are organized to form sentences. For his research, speech was recorded and then labeled manually for the experimental process. The system was trained and tested with the labeled speech and the final result achieved was 98.11%.

As we have seen from the above reviews, the earlier works for Afaan Oromo language in the area of speech recognition have the following characteristics: (i) only conducted for medium vocabulary size, and (ii) none of researchers conducted their research using broadcast news speech corpus. Thus, the aim of this research is to develop Afaan Oromo large vocabulary speech recognition system using broadcast news speech corpus.

# CHAPTER THREE

# AFAAN OROMO (THE OROMO LANGUAGE)

## 3.1. OVERVIEW OF AFAAN OROMO

Afaan Oromo belongs to the Cushitic branch of Afro-asiatic language family of languages of Africa which is figured to be divided into six major families like Chadic, Berber, Egyptian, Cushitic, Omotic, and Semitic. The Cushitic family is also further divided into four groups like North, Central, South and East. Accordingly, Afaan Oromo is one of the languages of Lowland groups within the East Cushitic group and the most widely spoken language from Cushitic family.

According to the Ethiopian census (2007), Afaan Oromo has about 34% speakers of the Ethiopian total population. Also Abraham (2014) stated that as Afaan Oromo was spoken by more than 40 million people over the world. And majority of these speakers of the language are living in Ethiopia and others are in neighboring countries like Kenya, Somalia, Egypt, and Djibouti.

Also Afaan Oromo is probably the third most widely spoken Afro-asiatic language in the world, after Arabic and Hausa (Teferi, 2015). And Abraham (2014) noted that Afaan Oromo is the 3rd largest language in Africa. From the Cushitic languages spoken in the Ethiopia, Afaan Oromo, Somali, Sidaama, Hadiya, and Afar-Saho are the languages which have the highest number of speakers (Abdella, 2010).

On the other hand, according to Tilahun (2004), about 95% of Afaan Oromo speakers live in Ethiopia, mostly in Oromiya region and also there are Oromo and non-Oromo people who speak Afaan Oromo as their second language. This is the reason that is possible to say definitely not less than two million non-Oromo speak Afaan Oromo as a second language. Additionally, people who are Oromo in their ethnic group are the first dominant speakers of this language as their mother tongue language. Therefore, Afaan Oromo is the working language of the Oromiya Regional State and used as a medium of instruction in both primary and junior secondary schools in Oromiya region and the language is currently taught as an examinable subject in the schools and also offered in some Ethiopian universities as a Minor and Major subject of study at both undergraduate and postgraduate level (Teferi, 2015).

## 3.2. AFAAN OROMO ALPHABETS "QUBEE"

Prior to 1991, Afaan Oromo does not have its own writing system but used Amharic characters called as Geez or Saban script as its writing system. However, since 1991 Afaan Oromo is written with a Latin alphabet called ***Qubee*** in Oromo, which was formally agreed in this year (Tilahun, 1993). Therefore, Afaan Oromo adopted the Latin letters for developing Alphabets that contain 32 letters. These letters have additional six double letters to 26 Latin letters.

Table 3.1 Category of Afaan Oromo Alphabets

| **Types** | **Afaan Oromo Alphabets** | **Total** |
|---|---|---|
| Vowel | A, E, I, O, U | 5 |
| Consonants | B, C, D, F, G, H, J, K, L, M, N, P, Q, R, S, T, V, W, X, Y, Z | 21 |
| | Compound symbol CH, DH, NY, PH, SH, TS | 6 |
| | | 32 |

The Afaan Oromo alphabets are basically classified into two main categories namely vowels ***'Qubee Dubbachiiftuu'*** and consonants ***'Qubee Dubbifamaa'.*** Vowels in Afaan Oromo are few in number (i.e., Afaan Oromo has five vowels). However, the number of consonants was more than the number of vowels. As represented in Table 3.1, from 32 alphabets 27 of them are consonants.

### 3.2.1. CATEGORIES OF AFAAN OROMO ALPHABETS

The Oromo alphabets can be classified based on their usage in constructing phone/syllable/word. Thus, the following discussion tells that types of Oromo alphabets.

#### i. Afaan Oromo Vowels (short vowels vs long vowels)

We have mentioned above as the vowels in Afaan Oromo are few in numbers. These vowels are used to make the sound with consonants and also alone by themselves. The sound made from these vowels classified into two namely short sounds when single vowel is used and long sounds when double similar vowel is used (Hinsene, 2009). Therefore, in Afaan Oromo vowels, sounds can be grouped into two by using a single vowel and doubling a similar vowel to make short sound and long sound, respectively.

For instance, let we see the Oromo words '**hara'** which means lake that have short sound and '**haaraa'** which means new that have long sounds.

### ii. Afaan Oromo Consonants (non-geminated vs geminated)

From Afaan Oromo alphabets consonants are larger than vowels in the language. In general speaking, all consonants cannot make the sound by themselves except they are preceded or followed by vowels. Accordingly, the created sound based on consonants also basically classified into two namely geminated and non-geminated.

In Afaan Oromo consonants, sounds can be grouped into two by using a single consonant and doubling a similar consonant to make non-geminated sound and geminated sound, respectively.

For instance, let us see the Oromo words '**haaduu**' which means shave that have non-geminated sound and '**haadduu**' which means blade that have geminated sounds. All Afaan Oromo consonants have geminate forms except *[h]* and compound symbols.

### iii. Compound symbol '*Qubee Dachaa*'

The compound symbol which is called "***qubee dachaa***" in Oromo is also categorized under consonants (Hinsene, 2009).This compound symbol is constructed in the language from double consonants to represent one single consonant.

Most scholars describe as there are only five compound symbols which are: CH, DH, NY, PH, and SH. But, as stated by Hinsene (2009), [TS] is not included in Afaan Oromo alphabets. In our work we consider that [TS] as one of Afaan Oromo alphabets and categorized under compound symbol. Because, from the data we have collected we have seen compound symbol [TS] which used especially in words like name of country and person.

For instance Egypt is written as '**Gibtsi**' in Afaan Oromo. In addition, from our audio data collected the speakers pronounce Egypt as '**Gibtsi**' in Afaan Oromo. In our study, we have used six compound symbols as shown in Table 3.1 previously. Therefore, **[TS]** can be one from Afaan Oromo compound symbols.

### iv. Afaan Oromo Glottal Sound

Afaan Oromo has a glottal sound which called ***'Hudhaa'*** that shows a diacritical marker and represented by single quote ['] or some times by **[h]**. For instance in the Oromo words like '***Ga'aa*** or ***Gahaa***' which means enough, both***'*** and ***h*** are used interchangeably and most of the time the single quote [‘] is used.

## 3.2.2. IPA REPRESENTATION OF AFAAN OROMO QUBEE

Afaan Oromo alphabets or letters have their International Phonetics Alphabet (IPA) representation. Still the language used Latin letters for developing its own writing system, but the IPA of these letters may differ from other language. Also while we compile the IPA for this language, we have considered the phones used in our transcription and the Table 3.2 showed IPA for Oromo alphabets.

Table 3.2 Afaan Oromo Alphabets and their pronunciation from IPA

| Alphabets/Qubee | Aa | Bb | Cc | CHch | Dd | DHdh | Ee | Ff | Gg | Hh | Ii |
|---|---|---|---|---|---|---|---|---|---|---|---|
| IPA for Qubee | [a] | [b] | [ʧ'] | [ʧ] | [d] | [ɗ] | [e] | [f] | [g] | [h] | [i] |
| **Alphabets/Qubee** | **Jj** | **Kk** | **Ll** | **Mm** | **Nn** | **NYny** | **Oo** | **Pp** | **PHph** | **Qq** | **Rr** |
| IPA for Qubee | [ʤ] | [k] | [l] | [m] | [n] | [ɲ] | [o] | [p] | [p'] | [k'] | [r] |
| **Alphabets/Qubee** | **Ss** | **SHsh** | **Tt** | **TSts** | **Uu** | **Vv** | **Ww** | **Xx** | **Yy** | **Zz** | **'** |
| IPA for Qubee | [s] | [ʃ] | [t] | [ ts] | [u] | [v] | [w] | [t'] | [j] | [z] | [ ʔ ] |

As it was presented in Table 3.2, the glottal symbol was considered as one letter and used to the glottal sound. We also discussed about glottal sound in our previous discussion.

## 3.3. MORPHOLOGICAL FEATURES OF AFAAN OROMO

Morphology is nothing but a branch of linguistic that studies and describes how words are formed in a language (Addunyaa, 2012). Morphology can be categorized into two categories of inflectional and derivational. Inflectional morphology is concerned with the inflectional changes in words where word stems are combined with grammatical markers for things like: person, gender, number, tense, case and mode. Inflectional changes do not result in changes of parts of speech. Derivational morphology deals with those changes that result in changing classes of words (changes in the part of speech).

Describing in detail about Afaan Oromo morphology was also not our main objective. But, we have discussed some of them in general in the next section. Thus, the following discussion deals about number and gender from morphological features of the language.

### i. Number

The number in Afaan Oromo shows the singular '**Qeentee'** and plural '**Heddumina'**. In addition, Afaan Oromo has different suffixes to form the plural of a noun and by nature most of nouns in the language were in their singular form (Addunyaa, 2012).

For instance, the Oromo word '**hiriyyaa**' which means friend and '**hriyyoota**' which means friends are singular and plural, respectively. From the example, the first word 'hiriyyaa' was singular without having any suffixes and affixes.

On the other hand, Afaan Oromo used the two numbering systems are used the two types of cardinal numbers and ordinal numbers which are used for counting and showing the order or rank in comparing, respectively which are illustrated in Table 3.3.

Table 3.3 Numbering system of Afaan Oromo for cardinal and ordinal numbers

| Type | English | Afaan Oromoo | Other name | Range |
|---|---|---|---|---|
| Cardinal Numbering | One | Tokko | One digit "Qub-tokkee" | 1 – 9 |
| | Two | Lama | | |
| | Three | Sadii | | |
| | . | . | | |
| | Nine | Sagal | | |
| | Ten | Kudhan | Two digits "Qublamee" | 10 – 99 |
| | Eleven | Kudha-tokko | | |
| | . | . | | |
| | Twenty | Digdama | | |
| | . | . | | |
| | Ninty nine | Sagaltamii-sagal | | |
| | Hundre | Dhibba | Three digits"Qub-sadee" | 100 – 999 |
| | Thousand | Kuma | Fourdigits"Qubarfee" | 1000 – 9999 |
| | Million | Miliyoona, Kitila | Six digits"Qub-jahee" | 1000000–999999 |
| | . | . | | . |

| | | | | |
|---|---|---|---|---|
| **Ordinal Numbering** | First | Tooffaa,Dursaa, Jalqaba | | |
| | Second | Lammata, Lammaaffaa, Lammeessoo, | | |
| | Third | Sadaffaa,Sadeessoo | | |
| | . | | | |
| | Tenth | Kurnaffaa, Kurneessoo | | |
| | . | . | | |

As we have seen from Table 3.3, the cardinal numbering system was used to count the amount of something and ordinal numbering system was used to show the order or rank. Both were used in our text transcribed and in the text corpus we have collected.

### ii. Gender

From Afaan Oromo morphological feature the other one is gender. The gender 'Koorniyaa' in Afaan Oromo is categorized into two as masculine 'Dhiira/Kormee' and feminine 'Dhalaa/Naayyee' (Addunyaa, 2012). Additionally, there are affixes which used to show the noun as they are masculine or feminine. For instance the Oromo word **Daa'ima** child we cannot simply identify the masculine or feminine from this word. However, by adding the Afaan Oromo affixes like '-icha and –ittii' on the word **Daa'ima**, we can show as Daa'imicha and Daa'imittii are the masculine and feminine, respectively. Therefore, from Afaan Oromo affixes '-icha' and '-ittii' are used to show the masculine and feminine of the noun in the language, respectively. There are also other affixes which are used for this purpose when they are added to the verb.

## 3.4. BASICS OF AFAAN OROMO PHONETICS

Afaan Oromo is spoken in the way it is written which help us to say the language is a phonetic language. This concept justified as Afaan Oromo is free from the problem of homonymy. This means there is no different words which are pronounced in the same way in Afaan Oromo like (write and right) in English. Even if the Latin letters were adopted for developing the Afaan Oromo alphabets, their pronunciation is different as the variety of language as we have discussed in Table 3.2. Also in Afaan Oromo alphabets, there is no difference while pronouncing the capital letters and small letters.

This means like the English language which used Latin letters, the pronunciations of **[A]** and **[a]** is same.

### 3.4.1. ARTICULATION OF AFAAN OROMO VOWELS

Phonemically, Afaan Oromo has five short and five long vowels which can occur word initially, medially or finally. It is recalled from the previous discussion of Section 3.2 as there are five vowels of Oromo Qubee. Also these vowels can be short and long as we have pointed previously. As Addunyaa (2014) discussed that, the pronunciations of these vowels were same overall the Oromiya region. These vowels have the following manner of articulating which presented in Table 3.4 below. There is no Afaan Oromo letter which can give the sound without the vowels.

Table 3.4 Articulation of Afaan Oromo Vowels (compiled from Addunyaa, 2014)

| Vowels | | | |
|---|---|---|---|
| | Front | Central | Back |
| Close | i /ɪ/, ii /iː/ | | u /ʊ/, uu /uː/ |
| Mid | e /ɛ/, ee /eː/ | | o /ɔ/, oo /oː/ |
| Open | | a /ʌ/ | aa /ɑː/ |

From the above table, front is the sound of a vowel formed by rising the tongue towards the hard plate and back is the sound of vowel articulated at the back of the mouth. However, only **[a]** is articulated at the central of our mouth. In general speaking the articulation Afaan Oromo alphabets (Qubee) vowels were based on the movement of the tongue (Addunyaa, 2014).

### 3.4.2. ARTICULATION OF AFAAN OROMO CONSONANTS

As we discussed before the most Afaan Oromo letters are categorized under consonants which means from 32 alphabets only five of them were vowels. In addition, these remaining 27 consonants have their pronunciation and manner of articulation. We have seen that from Table 3.4, the articulations of vowels were few that mean they can be articulated only at front, central, and back of our mouth and most vowels were close and mid whereas only [a, aa] was open.

In contrast, the articulation manner of consonants was differed from vowels articulation manner. As we have described in Table 3.5, articulation manner of consonants may nasal, stop, fricatives, and so on.

Additionally, in some sources the glottal sound symbol is categorized under consonants and we have discussed in earlier as it is represented by single quote [‘] or some times by letter [h]. However, in this section we elaborate on the pronunciation and the manner of articulation. Therefore, whether the symbol used for glottal sound was single quote or letter[h], the pronunciation representation from IPA is /ʔ/ , the manner of articulation is stops and fricatives, and glottal. In this section, we discuss mostly the manner of articulation for Afaan Oromo alphabets in Table 3.5.

Table 3.5 Articulation of Afaan Oromo Alphabets (compiled from Addunyaa, 2014).

| **Manner of Articulation** | **Place of Articulation** | | | | | |
|---|---|---|---|---|---|---|
| | **Bilabial** | **Labiodental** | **Alveolar** | **Palatal** | **Velar** | **Glottal** |
| Stop | b, ph | F | d, n | | k, g, q | ‘ /ʔ/ |
| Fricative | | | s, z | sh | | H |
| Affricative | | | | c, h, j | | |
| Nasal | M | | n | ny | | |
| Flap | | | r | | | |
| Lateral | | | l | | | |
| Semi –vowel | W | | | y | W | |

As it was presented in the Table 3.5, articulation of Afaan Oromo consonants depend on the human vocal bodies indication like mouth, lip, teeth, tongue and the like. Accordingly, manner of articulation of Afaan Oromo consonants may be stop, fricative, affricative, nasal, flap, lateral, and semi vowel; whereas place of articulation includes bilabial, labiodental, alveolar, palatal, velar, and glottal.

# CHAPTER FOUR

# CORPUS PREPARATION

## 4.1. INTRODUCTION

This chapter presents the several procedures that implemented in preparing corpus that we undertook in this research.

## 4.2. DATA PREPARATION

The first phase in developing speech recognition system is preparing the corpus required for the work. Therefore, our corpus preparation includes both text corpus and audio/speech corpus that we are going discuss.

### i. Text Corpus

The regularities in speech recognition which are used to estimate the probability of word sequence was taken or captured by language model. Language models are used to model regularities in natural language; also a list of large text was required in the development of language model. So, we have collected text data from ***Bariisaa*** Afaan Oromo newspaper and we have normalized manually these collected texts. Because, the text is by its nature contains different characters like, digits, dates, and abbreviation. In this study all numbers and abbreviation is putted in text form. In addition the written dates are transformed as they are spoken.

After pre-processing that collected text, we got 2000 Afaan Oromo sentences. These sentences were not much as we require and therefore we add the text that we have transcribed from the audio in order to increase the size of our text corpus.

### ii. Audio/Speech Corpus

Acoustic model is one from components of speech recognition. For modeling acoustic we need the audio/speech data. In other words, speech is one and the primary input for the recognizer system. We have used to ways for collecting speech data namely direct from ORTO studio and downloading from the websites of ORTO, FBC, and VOA. Their source and the way of collection were shown in Table 4.1.

Table 4.1 Proportion of speech data with respect to their sources

| Source | Way of collecting | NO. of utterance | Length | Sample rate |
|---|---|---|---|---|
| ORTO | From ORTO's studio | 2075 | 04:18:52 | 44.1 KHz |
| | From ORTO's website | 153 | 00:22:36 | |
| FBC | From FBC's website | 300 | 00:40:56 | |
| VOA | From VOA's website | 425 | 00:53:14 | |
| **Total Utterances** | | **2953** | **06:15:38** | |

As can be seen from Table 4.1, about 75% of the speech data was collected from ORTO and the remaining 25% of the speech collected from FBC. On the other hand, about 70% was collected directly from studio and remaining 30% from internet. Why we used the data from various source is to show the possibility of developing speech recognizer from different environment using broadcast news for Afaan Oromo.

### 4.2.1. AUDIO PRE-PROCESSING

The collected data need to be pre-processed, because all audio corpus gathered cannot be used in this work as it is. Additionally, the attention of this study was only on the continuous speech rather than spontaneous speech which consist of interviews. Therefore, all additional reports which include telephone reports and interviews were not used in this research. Accordingly, we conducted pre-processing on the collected audio of Afaan Oromo news in order to construct the speech data that is suitable format for HTK which described as below.

**Audio (speech) Segmentation:** Up on the knowledge of the researcher, there is no preferred tool to segment Afaan Oromo sentences automatically.

Therefore, sentence segmentation is done manually by listening audio file. During segmentation, telephone reports, interviews, and other non-speech like background music were skipped. Accordingly, the researcher constructed 2953 total speech utterances from 57 speakers (42 males and 15 females).

In addition, because HTK requires .WAV, .AIFF, and the like supported file formats, the researcher used open source software called Audacity 2.1.2 for preparing the audio file in sentence level and saving the audio files separately in **.wav** file format. The audacity is available on: http://audacity.sourceforge.net/download/windows.

The researcher does not recorded the speech from speakers. However, resampling the speech as required for this study in this limited resource was undertaken. So, while doing the audio segmentation we have saved segmented speech by using the default Audacity.

### 4.2.2. TRANSCRIBING THE SEGMENTED SPEECH

We said that as the researcher constructed 2953 total speech utterances from 57 speakers. After that each and every sentence should be represented with correspondence transcriptions. But, there is no appropriate tool at hand that assists us to produce text format from audio speech. Therefore, the transcription was done manually by the researcher for each sentence.

Furthermore, the task of transcription was accomplished by considering grammar of Afaan Oromo and consulting different literatures on Afaan Oromo word construction. As we have discussed earlier in chapter three, the language has glottal sound which is called and represented by single quote or apostrophe [’] and sometimes by letter [h] (Hinsene, 2009). However, different scholars stated different ideas to categorize this glottal sound whether it is consonant or vowel. Thus most of them argued that it is consonant and serve as consonant according to the grammar of the language and some of them argued that it is classified as a vowel. But for this study, we have decided that to agree with the glottal symbol is used as consonant and we count it as one from Afaan Oromo consonant alphabets.

As mentioned prior in Table 3.2, the IPA of Oromo alphabets are given. From that IPA representation of some letters like C, CH, DH, J, NY, PH, Q, SH, X are non ASCII code which HTK cannot support. Additionally, the symbol of glottal sound itself is not represented by ASCII code and this make challenges to the researcher while making the transcription. Therefore, we are obliged to modify the IPA for such letters and the glottal symbol in this research. As can be seen from Table 3.2, most of the letters and their IPA representations were similar. So, we have used same fashions for the remaining letters like C, CH, DH, J, NY, PH, Q, SH, X. This means, letters were corresponding to their IPA representations in this research as shown in Table 4.2.

By referring different literatures on phonics of language, we have used [hh] for representing glottal sound. The reason that we used [hh] is, the linguists of Afaan Oromo

were stated that [h] cannot be geminated and sometimes [h] is used instead of glottal sound in some words (Tilahun, 1993).

For instance in Oromo words like ***gahaa*** and ***ga'aa*** which means enough, both [h] and [ʼ] were used interchangeably. After studying about all phones of the language in detail, and based on the ideas of those linguists, the researcher used [hh] to represent the glottal symbol for this particular research. Because, there is no [hh] phones in the language and it is recalled from chapter three that all consonants can have geminated and non-geminated form except [h]and compound symbols ***'Qubee Dachaa'***. Double consonants are used to represent the geminated form.

Table 4.2 Afaan Oromo phones used during transcription

| Letters | A | AA | B | BB | C | CC | CH | D | DD | DH | E | EE | F | FF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **IPA** | **A** | **Aa** | **b** | **bb** | **C** | **Cc** | **Ch** | **D** | **dd** | **dh** | **e** | **Ee** | **f** | **ff** |
| Letters | G | GG | H | I | II | J | JJ | K | KK | L | LL | M | MM | N |
| **IPA** | **G** | **Gg** | **h** | **I** | **Ii** | **J** | **Jj** | **K** | **kk** | **L** | **ll** | **M** | **mm** | **N** |
| Letters | NN | NY | O | OO | P | PP | PH | Q | QQ | R | RR | S | SS | SH |
| **IPA** | **Nn** | **Ny** | **o** | **oo** | **P** | **Pp** | **Ph** | **Q** | **qq** | **R** | **rr** | **S** | **ss** | **sh** |
| Letters | T | TT | TS | U | UU | V | W | WW | X | XX | Y | YY | Z | ʻ |
| **IPA** | **T** | **Tt** | **ts** | **U** | **Uu** | **V** | **W** | **ww** | **x** | **xx** | **y** | **yy** | **z** | **hh** |

By doing so, the researcher transcribed 2953 Afaan Oromo utterances for the experiment. While doing this the verity of dialects, punctuations, and the rule of capitalization were not considered. In other words, all transcribed were in lower case and without punctuations.

## 4.3. HIDDEN MARKOV MODELS TOOLKIT (HTK)

We have used HTK which is a toolkit for building Hidden Markov Models (HMMs). Young et. al. (2006) stated that as HTK is primarily designed for building HMM based speech processing tools, particularly speech recognizers. Also much of the functionality of HTK is built into the library modules. The figure 4.1 shows the HTK software architecture. Figure 4.1 illustrates the software structure of the HTK tool and shows its input/output interfaces. User input/output and interaction with the operating system is controlled by the library module HShell and all memory management is controlled by HMem.

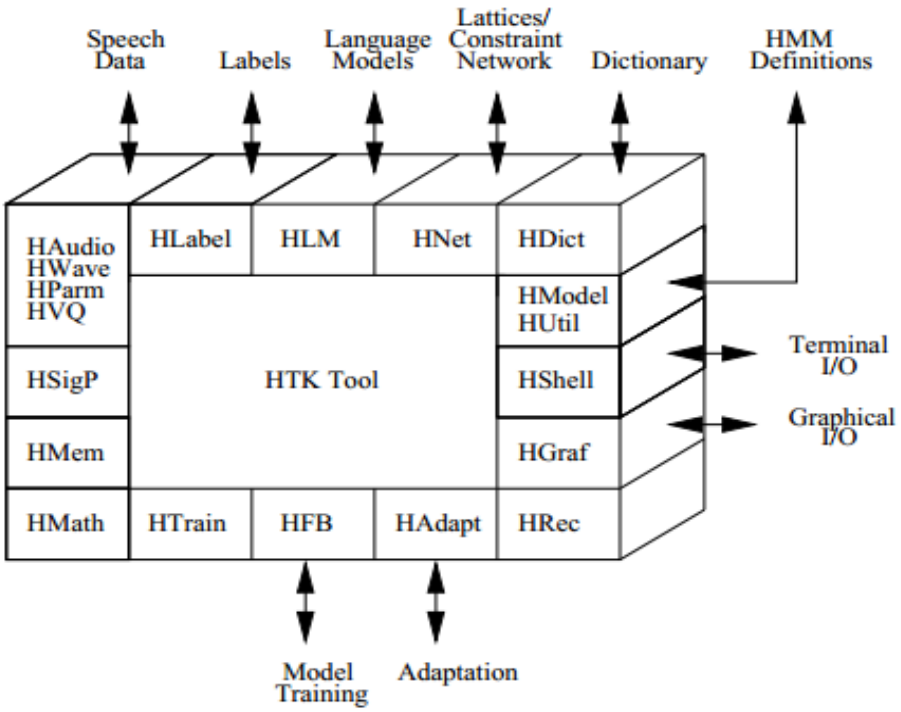


Figure 4.1 Software Architecture (taken from Young et.al, 2006)

Math support is provided by HM at hand the signal processing operations needed for speech analysis are in HSigP. Each of the file types required by HTK has a dedicated interface module. HLabel provides the interface for label files, HLM for language model files, HNet for networks and lattices, HDict for dictionaries, HVQ for Vector Quantisation (VQ) codebooks and HModel for HMM definitions. In the next section we discussed the tools taken from HTK and that we have used for our work as they were required.

### i. Data Preparation Tool

Data preparation is the first step in the development of speech recognition system as several researchers do that. So to build a set of HMMs, a set of speech data files and their associated transcriptions are needed. Because before bringing speech data and using in training, it must be converted into the required correct format and phone or word labels (Young et. al. 2006). While all HTK tools can parameterise waveforms on-the-fly, in practice we need to parameterise the data just once. So the tool **HCopy** is used for this purpose. Also the tool **HList** is used to check the contents of any speech file and simply it can be used to check the result of the conversions before using. In order to output the files to a single Master Label File (MLF), the tool **HLEd** was used.

### ii. Training Tools

Defining the topology required for each HMM by writing a prototype definition is the second step of system building. Also HTK allows HMMs to be built with any desired topology. HMM definitions can be stored externally as simple text files and hence it is possible to edit them with any convenient text editor. The purpose of the prototype definition is only to specify the overall characteristics and topology of the HMM. In the study all of the phone models are initialised to be identical and have state means and variances equal to the global speech mean and variance because of no bootstrap data is available. The tool HCompV can be used for this purpose.

### iii. Recognition Tools

HTK provides a single recognition tool called HVite which uses the token passing algorithm to perform Viterbi-based speech recognition (Young et. al. 2006). 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 be performed on either a list of stored speech or on direct audio input. However, in this study we have used HVite which can support triphones and monophones that can run with multiple tokens to generate lattices containing multiple hypotheses. It can also be configured to rescore lattices and perform forced alignments. The word networks needed to drive HVite are usually either simple word loops in which any word can follow any other word or they are directed graphs representing a finite-state task grammar. In the former case, bigram probabilities are normally attached to the word transitions.

### iv. Analysis Tool

Once the HMM-based recogniser has been built, it is necessary to evaluate its performance. This is usually done by using it to transcribe some pre-recorded test sentences and match the recogniser 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.

# CHAPTER FIVE

# EXPERIMENT AND RESULTS

## 5.1. INTRODUCTION

The steps of HTK are broadly categorized into four phases: data preparation phase, training phases, recognition phase, and analysis phase. Therefore, in this chapter we discussed our experiments that conducted through these phases.

## 5.2. DATA PREPARATION PHASE

Data preparation is the first step in the development of speech recognition system as several researchers do that. So to build a set of HMMs, a set of speech data files and their associated transcriptions are needed. It is recalled that we have described about data preparation in chapter 4. The data preparation phase can be accomplished through a several steps like: constructing pronunciation dictionary, creating the transcription files, and coding the audio data.

### 5.2.1. PRONUNCIATION DICTIONARY

Most of the time, creating a sorted list of the words that contains in the grammar one per line with their pronunciations is the initial step in building the Pronunciation Dictionary. Therefore, to create a pronunciation dictionary we need to have list of words (**`trainwordlist`** in our context) which is a sorted list of the unique words that appear in the training transcription. For this study, we derived a list of unique words from sentences using the Perl script **prompts2wlist** that invoked as follow to create **`trainwordlist`** and **`testwordlist`**, which contains 10138 and 2516 unique words, respectively.

```
perl script/prompts2wlist trainprompt.txttrainwordlist
perl script/prompts2wlist testprompt.txttestwordlist
```

All sentences that we have used was saved as **`trainprompt.txt`** and **`testprompt.txt`** files that contained sentences prepared for training and testing, respectively. Samples of **`trainprompt.txt`** and **`testprompt.txt`** are attached in Appendix B.

Pronunciations are the phonemes that make up a word. Also the dictionary can provides a connotation between words used in the task grammar and the acoustic models which composed of sub word. It can be in (phonetic, syllabic, and etc.) units. In this study we have constructed phone based pronunciation because the phone of the language is fixed. Hence, using 56 Afaan Oromo phones the pronunciation dictionary was constructed manually with file name **`traindictionary.txt`** for this study. These 56 phones of the language were derived from 10 vowels (5 short and 5 long), 6 compound symbols, and 40 consonants (21 non-geminated, 18 geminated and 1 glottal symbol).

We have wrote the phones with their IPA for making our constructed dictionary standard. However, there was an exception since all IPA cannot be written in ASCII code. Therefore, correction was done manually for difficulties of such letters (i.e., their IPA was difficult to put with their ASCII code) and phones used for this purpose were presented in Table 4.2.

The **`HDMan`** command was used to go through the list of words (**`trainwordlist`**) file and look up the pronunciation for each word from a separate pronunciation files that we prepared and output the result in a **`traindictionary`**.

```
HDMan -m -w trainwordlist -n trainmonophones1 -l traindlog traindict
traindictionary.txt
```

From the above command the following three files created:

a. ***`trainmonophones1`****:* this includes a list of the phones used in our dictionary.
b. ***`traindict:`*** the pronunciation dictionary for our Grammar and additional words required to create a phonetically balanced Acoustic Model; and
c. ***`traindlog`*****:** which shows the frequency of phones used in the dictionary

A new dictionary called **traindict** is created by searching the source dictionary ***`traindictionary.txt`*** we have developed for this purpose to find pronunciations for each word in **`trainwordlist`**. The general format of each dictionary entry created by HDMan command is: **`word [word] p1 p2 p3`** … That means that the word is pronounced as the sequence of phones p1 p2 p3 … The string in square brackets [word] specifies the string to output when that word is recognized. If it is omitted then the word itself is output. If it is included but empty, then nothing is output.

For instance, when the Oromo word '**aadaa**' which means culture is written by the above format as: **aadaa [aadaa] aa d aa**

The vocabulary words used for training in this experiment constructed from 56 Afaan Oromo phones without including sp and sil. Also the dictionary built for this thesis was alternative dictionary, because it includes words which have more than one pronunciation. For instance the word `itiyoophiyaa` pronounced as: `itiyoophiyaa, itoophiyaa,` and `xoophiyaa.` The sample of our pronunciation dictionary attached in Appendix C.

While constructing the alternative pronunciation dictionary, we put at first the standard one and writing in different ways of they pronounced. Accordingly, for the above mentioned example;

```
itiyoophiyaa itiyoophiyaa  i t i y oo ph i y aa sp
             itoophiyaa    i t oo ph i y aa sp
             toophiyaa     t oo ph i y aa sp
             xoophiyaa     x oo ph i y aa sp
```

Some words with their frequency occurred in our alternative dictionary was summarized in Appendix D.

Generally, our train pronunciation dictionary contain 10,247 words and all 56 phones were covered. However, the size of pronunciation dictionary **traindict** was 10140 and 58 phones were covered after running **HDMan**.

**Creating Mono-phones**

We simply create the mono-phones (**trainmonophones0** in our case) from the above created **trainmonophones1** by executing **HDMan** command and by removing sp from **trainmonophones1**.

### 5.2.2. CREATING THE TRANSCRIPTION FILES

Before bringing speech data and using in training, it must be converted into the required correct format and phone or word labels.

The process of transcription can be performed as word level transcription or phone level transcription.

### i. Word Level Transcriptions

Since HTK toolkit cannot process sentences file directly, for both sets of phone label and word level transcriptions, an orthographic transcription in HTK Label format is needed. To do this we have two options. The first option we can create a separate label file for each line of our sentence file using a text editor. The second option, we can create a Master Label File (MLF) using scripting language. MLF file is a single file that contains a label entry for each line in prompts file. Since the second one is the easiest approach we have preferred it for our experiment. In order to generate the .mlf file, we have used the Perl script **prompts2mlf**which is provided with HTK samples.

```
perl scripts/prompts2mlf trainwords.mlf trainprompts
```

After the above command executed, the word level transcription, **trainwords.mlf** file was created.

### ii. Phone Level Transcriptions

The HLEd command which provided with HTK tool was executed after word level transcription to expand the Word Level Transcriptions to Phone Level Transcriptions. By this command each word is replaced by its equivalent phonemes and put the result in a new phone level master label file. This is done by reviewing each word in the MLF file, and looking up the phones that make up that word in the dictionary file we created earlier, and out putting the result in a file called **trainmophones0.mlf** which do not have short pauses (sp) after each word phone group.

Before executing this command first, we have created the **mkphones0.led** edit script with the following content:

```
EX
IS silsil
DE sp
```

Then executing the **HLEd** command creates the **trainphones0.mlf** file.

```
HLEd -A -D -T 1 -l * -d traindictionary.txt –I trainphones0.mlf
mkphones0.led trainwords.mlf
```

Next, the second **trainphones1.mlf** file which includes short pause (sp) after each word phone group created. In this case before the execution of the **HLEd** command, we have also created the**m kphones1.led** edit script with the following contents:

```
EX
IS sil sil
```

Then executing the **HLEd** command creates **trainphones1.mlf**filefor us.

```
HLEd -A -D -T 1 -l * -d traindictionary.txt –I trainphones1.mlf
mkphones1.ledtrainwords.mlf
```

### 5.2.3. FEATURE EXTRACTION

While all HTK tools can parameterise waveforms on-the-fly, in practice we need to parameterise the data just once and the tool **HCopy** is used for this purpose. In order to output the files to a single Master Label File (MLF), the tool **HLEd** was used.

The final stage of data preparation is to parameterize the raw speech waveforms into sequences of feature vectors (Young et.al, 2006). Because of HTK is not efficient in processing wav files as it is with its internal format, we need to convert our audio wav files to another format.

HTK support both FFT-based and LPC-based analysis. Here we have used Mel Frequency Cepstral Coefficients (MFCCs), which are derived from FFT-based log spectra.

We used the **HCopy** tool to convert our wav files to MFCC format. For doing this, we have 2 options; one option we can execute the **HCopy** command by hand for each of our audio (wav) files, or in other option we can create a file containing a list of each source audio file and the name of the MFCC file it will be converted to, and use that file as a parameter to the **HCopy** command. We will use the second one in this experiment. To do this we have created the **traincode.scp** script file that contained all list of utterances used in the training and used to extract features of speech and create the mfcc files for each utterance. The **traincode.scp** and **testcode.scp** script files are written as below and for more samples see at Appendix E.

```
train/audio/onew1.wav      train/mfcc/onew1.mfc
train/audio/onew2.wav      train/mfcc/onew2.mfc
```

```
train/audio/onew3.wav      train/mfcc/onew3.mfc
train/audio/onew4.wav      train/mfcc/onew4.mfc
train/audio/onew5.wav      train/mfcc/onew5.mfc
        …                          …
```

**HCopy -A -D -T 1 –C config_hcopy -S traincode.scp**

Then the **HCopy** command performs the conversion from wav format to MFCC and a configuration file which specifies all the needed conversion parameters is required. Thus, for this purpose we have prepared the configuration file (config_hcopy) with different parameters. These different parameters are elaborated as follow:

```
TARGETKIND = MFCC_0_D_A    #Identifier of the coefficients
SOURCERATE = 625.0         # = 16 kHz shows the resampled
SOURCEKIND = WAV           # Identifies as source kind is wave
SOURCEFORMAT = WAV         # Source format used is wave
TARGETFORMAT = HTK         # Target format is
HTKENORMALISE = F          # no energy normalization performs
TARGETRATE = 100000.0      # = 10 ms = frame periodicity
WINDOWSIZE = 250000.0      # = 25 ms = length of time frame
USEHAMMING = T             # Use of Hamming windowing function
PREEMCOEF = 0.97           # Pre-emphasis coefficient
NUMCHANS = 26              # Number of filter-bank channels
CEPLIFTER = 22             # Length of cepstral liftering
NUMCEPS = 12               # Number of MFCC coeffs
```

Executing HCopy command as follows creates a series of mfc files corresponding to the audio files as it is listed in **traincode.scp**.

**HCopy -A -D -T 1 -C config_hcopy –S traincode.scp**

Different configuration files are given under Appendix F that we have been used for this study.

### 5.3. TRAINING PHASE

Defining the topology required for each HMM by writing a prototype definition is the second step of system building. Also HTK allows HMMs to be built with any desired topology. HMM definitions can be stored externally as simple text files and hence it is possible to edit them with any convenient text editor. The purpose of the prototype definition is only to specify the overall characteristics and topology of the HMM. In our

study all of the phone models are initialised to be identical and have state means and variances equal to the global speech mean and variance because of no bootstrap data is available. The tool **HCompV** can be used for this purpose.

**Training the Model**

Training is the next step which can be undertaken after the work of preparing the required data for training. Also splitting our data for training and for testing is another important task which must be accomplished at this level. Table 5.1 shows the proportion of our prepared data as their usage for training and testing purpose.

Table 5.1 Proportion of our prepared speech data for training and testing

| **Data Used** | **NO. of Sentences** | **Length/ Duration** | **NO. of Speaker** | **Speaker Gender** | |
|---|---|---|---|---|---|
| | | | | **Male** | **Female** |
| Training | 2653 | 05:34:42 | 45 | 33 | 12 |
| Testing | 300 | 00:40:56 | 12 | 9 | 3 |
| ***Total*** | ***2953*** | ***06:15:38*** | ***57*** | ***42*** | ***15*** |

As can be seen from Table 5.1, totally 2953 utterances (06:15:38 hours long) was used for the experiment of this research. Accordingly, out of 2953 utterances, 2653 of the utterances (05:34:42 hours of speech) were used to train the model and the remaining 300 utterances which collected form FBC (00:40:56 minutes of speech) were used for testing the developed speech recognizer system.

### 5.3.1. CREATING MONO-PHONE AND TRI-PHONE HMMS

**Prototype Definition**

The principal step of an HMM model training is defining the prototype model. As Young et. al. (2006) stated that, the parameters of this model are not important; its purpose is to define the model topology. Hence, our recognition system is phone-based system, and we have done training with left to right HMM topology of 5-state (3 emitting states and two non-emitting) left to right without skipping. To define the topology we have created the file proto, which is in appendix G.

The HTK tool **HCompV** will 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.

```
HCompV -A -D -T 1 -C config_hcompv -f 0.01 -m -S train.scp -M hmm0
proto
```

Therefore, supposing that a list of all the training files is stored in `train.scp`, the above command creates the new version of our proto file in which the zero means and unit variances are replaced by the global speech means and variances.

Using new prototype model generated by **HCompV**, a Master Macro File (MMF) called **hmmdefs** containing a copy for each of the required mono-phone HMMs is constructed manually by copying the prototype and relabeling it for each required mono-phone including "sil". Consequently, the file macros contain a global options macro and the variance floor macro vFloors generated earlier by HCompV. The global options macro simply defines the HMM parameter kind and the vector size.

### 5.3.2. RE-ESTIMATING MONO-PHONES

The mono-phones created are re-estimated using the embedded re-estimation tool HERest invoked as:

```
HERest -C config_hcompv -I trainphones0.mlf -t 250.0 150.0 1000.0 -S
train.scp -H hmm0/macros -H hmm0/hmmdefs -M hmm1 trainmonophones0
```

Then the above command loads all the models both hmmdefs and macros which are listed in the model list. Mono-phones used here are excluding the short pause (sp) model. These are then re-estimated using the data listed in `train.scp` and the new model set is stored. In the command, $-t$ option sets the pruning thresholds to be used during training and this pruning is normally 250.0. If re-estimation fails on any particular file, the threshold is increased by 150.0 and the file is reprocessed. This is repeated until either the file is successfully processed or the pruning limit of 1000.0 is exceeded.

#### Fixing the silence

Silence models has been created by running the HMM editor HHEd to add the extra transitions required and tie the sp state to the centre sil state HHEd works in a similar way to HLEd. It applies a set of commands in a script to modify a set of HMMs. In this case, it is executed by the following command.

```
HHEd -H hmm4/macros -H hmm4/hmmdefs -M hmm5 sil.hed trainmonophones1
```

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]}
```

At this point, 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 file and within each silence model, the original state parameters are replaced by the name of this macro. The new list of mono-phones contains sp model is used in the above HHEd command. At the end, HERest are applied using the phone transcriptions with sp models between words.

**Realigning the Training Data**

We have said earlier as our pronunciation dictionary was alternative pronunciation dictionary. So, 109 words have at least two pronunciations from our train pronunciation dictionary. For this reason we need to realign the trained data. The basic difference between realigning the training data and the original word to-phone mapping performed by HLEd in data preparation is that, in the operation of realigning all pronunciations for each word are considered and outputs the pronunciation that best matches the acoustic data. Using the phone models created before we have realigned the training data then creates new transcriptions with a single call of the HTK recognition tool HVite as follows:

```
HVite -l * -o SWT -b sent-end -C config_hvite -a -H hmm7/macros -H
hmm7/hmmdefs –i aligned.mlf -m -t 250.0 -y lab -I oromowords.mlf -S
train.scp traindictionary.txt trainmonophones1>HVite_log
```

This command uses the HMMs made previously to transform the input word level transcription **trainwords.mlf** to the new phone level transcription **aligned.mlf** using the pronunciations stored in the traindictionary.txt that constructed so far. When aligning the data, it is sometimes clear that there are significant amounts of silence at the beginning and end of some utterances, so to spot this the time-stamp information need to be output during the alignment. That is why we have used the option -o SW in the above command. After the new phone alignments have been created, HERest applied to re-estimate the HMM set parameters.

### 5.3.3. REFINEMENTS AND OPTIMIZATION

#### Tied-State Tri-phones

As pointed by Young et. al. (2006), the first stage of model refinement is usually to convert a set of initialized and trained context-independent mono-phone HMMs to a set of context dependent models. But before building a set of context-dependent models, it is necessary to decide whether or not cross-word tri-phones are to be used. If they are context dependent, then word boundaries in the training data can be ignored and all mono-phone labels can be converted to tri-phones. If word internal tri-phones are to be used, then word boundaries in the training transcriptions must be marked. So, we have built context dependent tri-phones for this study.

Since we have prepared a set of mono-phone HMMs with the previous steps, now we can use them to create context-dependent tri-phone HMMs. We did this in two steps. Firstly, the mono phone transcriptions are converted to tri-phone transcriptions and a set of tri-phone models are created and re-estimated. Secondly, similar acoustic states of these tri-phones are tied. Tying is nothing but it is the method of making one or more HMMs share the same set of parameters. Then the set of context-dependent models re-estimated using HERest tool.

#### Making Tri-phones from Mono-phones

Context-dependent tri-phones can be created from mono-phones. The tri-phone transcription have be created first using HLEd tool which enables us to generate a list of all the tri-phones for which there is at least one example in the training data.

```
HLEd-n triphones1 –l * –I wintri.mlf mktri.led aligned.mlf
```

Using the above command we have created the tri-phone transcriptions in `wintri.mlf` file using mono-phone transcriptions `aligned.mlf` file. At the same time, a list of tri-phones is written to the file `triphones1`. The edit script `mktri.led` used above contains the commands:

```
WB sp
WB sil
TC
```

The two WB command defines sp and sil as word boundary symbols. These then blocks the addition of context in the TI command, which converts all phones except word boundary symbols to tri-phones. For example,

**`sil a kk a m sp`** becomes **`sil a+kk a-kk+a kk-a+m a-m sp`**

This tri-phone transcription is word internal. Some bi-phones may be generated as contexts at word boundaries since sometimes they include only two phones.

```
HHEd -B -H hmm9/macros -H hmm9/hmmdefs -M hmm10 mktri.hed
trainmonophones1
```

Where the edit script mktri.hed contains a clone command CL followed by TI commands to tie all of the transition matrices in each triphone set, that is:

```
CL triphones1
     TI T_aa {(*-aa+*,aa+*,*-aa).transP}
     TI T_d {(*-d+*,d+*,*-d).transP}
     TI T_sp {(*-sp+*,sp+*,*-sp).transP}
     TI T_n {(*-n+*,n+*,*-n).transP}
     TI T_t {(*-t+*,t+*,*-t).transP}
```

We have generated the file mktri.hed using the Perl script maketrihed included in the HTK Tutorial directory. The clone command CL takes as its argument the name of the file containing the list of triphones (and biphones) generated above.

For each model of the form `a-b+c` in this list, it looks for the monophone b and makes a copy of it. Due to we use latter the transition matrix `transP` which is regarded as a sub-component of each HMM, each TI command takes as its argument the name of a macro and a list of HMM components. The lists of items within brackets are patterns designed to match the set of tri-phones, right bi-phones and left bi-phones for each phone.

**Making Tied State Tri-phone**

After a set of tri-phone HMMs with all tri-phones in a phone set sharing the same transition matrix prepared now we can tie them. Tying states within tri-phone sets helps to share data and thus be able to make robust parameter estimates. HTK tool 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 tri-phone. The decision tree attempts to find those contexts which make the largest difference to the acoustics and which should therefore distinguish clusters. Decision tree state tying is performed by running HHEd:

```
HHEd -B -H hmm12/macros -H hmm12/hmmdefs -M hmm13 tree.hed triphones1
```

The edit script `tree.hed` contains the instructions regarding which contexts to examine for possible clustering and the detail contents `tree.hed` was attached at appendix H.

We have used mkclscript script which is found in the RM Demofor creating the TB commands (decision tree clustering of states) which is one part of tree.hed. Firstly, the RO command is used to set the outlier threshold to 100.0 and load the statistics file generated at the end of the previous step. The outlier threshold determines the minimum occupancy of any cluster and prevents a single outlier state forming a singleton cluster just because it is acoustically very different to all the other states. The TR command sets the trace level to zero in preparation for loading in the questions.

Each QS command loads a single question and each question is defined by a set of contexts. For example, one of the QS command defines a question called 'R_Nasal' which is true if the right context is either of the nasals n, ny, or m n tree.hed file using QS command. So, the questions referring to both the right and left contexts of a phone are included. The full set of questions loaded using the QS command would include every possible context which can influence the acoustic realization of a phone, and can include any linguistic or phonetic classification which may be relevant. For this study we have constructed the questions (QS) which are attached at Appendix H.

The set of tri-phones used so far only includes those needed to cover the training data. The AU command takes as its argument a new list of tri-phones expanded to include all needed for recognition. This list can be generated, for example, by using HDMan on the entire dictionary (not just the training dictionary), converting it to tri-phones using the command TC and outputting a list of the distinct tri-phones to a file using the option –n.

```
HDMan -b sp -n fulllist -g global.ded -l flog dict-tri
traindictionary.txt
```

The -b sp option specifies that the sp phone is used as a word boundary and it is excluded from `triphones`. The effect of the AU command is to use the decision trees to synthesize all of the new previously unseen triphones in the new list.

Once all state-tying has been completed and new models synthesized, some models may share exactly the same states and transition matrices and they became identical. The CO command is used to compact the model set by finding all identical models and tying them together, producing a new list of models called `tiedlist`. One of the advantages of using decision tree clustering is that it allows previously unseen tri-phones to be synthesized. To do this, the trees must be saved and this is done by the ST command. Finally, the models are re-estimated using HERest.

**Increasing Gaussian mixture**

We have constructed mono-phones and context dependent tri-phones in the previous stages with only single Gaussian models. However, the increment of Gaussian mixture has weighty effect on the performance of recognizer. It is better to increase the Gaussian mixture of the models we have constructed in previous tasks. By doing so the Gaussian Mixtures was increased to 12 in this study to achieve on best performance.

According to Young et.al. (2006), in HTK the conversion from single Gaussian HMMs to multiple mixture component HMMs is usually one of the tasks in a system refinement. The mechanism provided to do this is the HHEd tool MU command which helps to increase the number of components in a mixture by a process called mixture splitting. We used this approach to building a multiple mixture component system which is particularly flexible since it allows the number of mixture components to be repeatedly increased until the desired level of performance is achieved. For this purpose, the MU command was used and the MU command has the form: *MU* $\boldsymbol{n}$ *itemLists*

Where $\boldsymbol{n}$ is the new number of mixture components required and *itemLists* defines the actual mixture distributions which required to modify. For instance, to increases the number of mixture components in the output distribution for state 2 to 4 of all models to 2 is defined as the following MU command:

MU 2 {*.state[2-4].mix}

**Language Model**

The Language model used for recognition was developed using 2000 sentences text corpus taken from Bariisaa Afaan Oromo Newspaper. Since this text corpus was small, we add the text corpus that we have prepared during transcription. The reason we have done this is to increase the size of our data. Because, when the size of text data increased, the probability of occurrence of words also increased. Then the increment of word probability has an improvement on the accuracy of speech recognizer. We have used **SRILM** language modelling tool for development of bigram language model and trigram language model. The **HTK** language modelling tool we have used to build bigram language model and word network were **HLStats** and **HBuild**, respectively. Using these two different language modelling tools was for only identifying which performs better. We also done the experiment with closed vocabulary, pronunciation dictionary.

## 5.4. RECOGNITION PHASE

During recognition phase, HTK provides a single recognition tool called **`HVite`** which uses the token passing algorithm to perform Viterbi-based speech recognition. **`HVite`** takes as input a network describing the allowable word sequences, a dictionary defining how each word is pronounced and a set of HMMs

The word networks needed to drive HVite are usually either simple word loops in which any word can follow any other word or they are directed graphs representing a finite-state task grammar. In the former case, bigram probabilities are normally attached to the word transitions.

**Recognizing**

The task of searching for the most likely sequence of words given in observed features extracted from the speech signal is usually referred to as decoding or recognizing the speech signal.

Decoding speech was begun by constructing a search graph which contains every word in the recognition vocabulary. Each word is then replaced by the HMMs that correspond to the sequence of sound units which make up the word. As a result, the search graph is a large HMM, and recognition is performed using the Viterbi algorithm to

align the search graph to the speech features derived from the utterances. Because the Viterbi algorithm is used to find the most likely word sequence, the decoding procedure is said to be done via Viterbi search.

Suppose that the ***test.scp*** file holds a list of the coded test files, and then each test file recognized and its transcription output to an MLF file called ***recout.mlf*** by executing the following:

```
HVite -H hmm15/macros -H hmm15/hmmdefs -S test.scp -l * -i
recout.mlf -w wordnetwork -p 1.0 -s 9.0 traindict tiedlist
```

The options -p and -s set the word insertion penalty and the grammar scale factor, respectively. The word insertion penalty is a fixed value added to each token when it transits from the end of one word to the start of the next. The grammar scale factor is the amount by which the language model probability is scaled before being added to each token as it transits from the end of one word to the start of the next. Because of these parameters can have a significant effect on recognition performance, we have made some modification and conducted the experiment with different parameters.

### 5.5. ANALYSIS PHASE

The final stage of the HTK toolkit is the analysis stage. Once the HMM-based recogniser has been built, it is necessary to evaluate its performance. This is usually done by using it to transcribe pre-recorded test sentences and match the recogniser 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.

Once the test data has been processed by the recognizer, the next step is to analyse the results. The HTK tool **`HResults`** is provided for this purpose. **`HResults`** compares the transcriptions output by HVITE with the original reference transcriptions and then outputs various statistics. **`HResults`** matches each of the recognized and reference label sequences by performing an optimal string match using dynamic programming. Once it founds the optimal alignment, **`HResults`** calculates the number of substitution errors (S), deletion errors (D) and insertion errors (I).

Then outputs both percentage of correctly recognized words and percentage of accurately recognized words. Since HTK tools can process individual label files and files stored in MLFs, we have used our MLF file which contains word level transcriptions for test file called **testref.mlf**, the actual performance is determined by running HResults command as follows:

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

After running HResults command there are several results that we have seen. So, for the finding of this study we tried to describe in following tables. Our experiment was conducted using different parameters for word insertion penalty and grammar factor scale. In this experiment, we used test data that collected only from FBC, and the default value of Gaussian Mixture was used. Accordingly, the result of our experiments was presented in Table 5.2.

Table 5.2 Result of experiments conducted by tuning parameters for –p and –s options

| **Parameter tuning Values** | | **Per cent of Word** | | **Word Error Rate (%WER)** |
|---|---|---|---|---|
| **– $p$ options** | **– $s$ options** | **Correctly Recognized** | **Accurately Recognized** | |
| 0.0 | 5.0 | 13.23 | 2.41 | 97.59 |
| 0.2 | 7.0 | 9.62 | 4.18 | 95.82 |
| 0.4 | 9.0 | 12.31 | 4.28 | 95.72 |
| 0.6 | 11.0 | 10.90 | 6.65 | 93.35 |
| 0.8 | 13.0 | 11.63 | 8.95 | 91.05 |
| **1.0** | **15.0** | **12.31** | **9.20** | **90.81** |

As can be seen from Table 5.2, the best performance was achieved when tuning the parameters to 1.0 and 15.0 for word insertion penalty and grammar scale factor, respectively. The small word error rate obtained that has the best performance from our experiments was 90.81% WER, based on different parameters of word insertion penalty and grammar scale factor.

We have also conducted another experiment by having the above values for –p and –s options and bay changing the values of Gaussian Mixtures. Because, increasing Gaussian Mixture is one technique for improving the performance of the recognizer. Accordingly, the result obtained during these experiments were presented in Table 5.3. The change during in this experiment was only changing the number of Gaussians.

Table 5.3 Result of experiments conducted by increasing Gaussian Mixtures

| Number of Gaussians | Per cent of Word | | Word Error Rate (%WER) |
|---|---|---|---|
| | Correctly Recognized | Accurately Recognized | |
| 2 | 12.41 | 4.01 | 95.99 |
| 4 | 12.31 | 4.28 | 95.72 |
| 6 | 11.63 | 6.65 | 9335 |
| 8 | 10.89 | 8.89 | 91.13 |
| 10 | 10.90 | 9.20 | 90.81 |
| **12** | **11.90** | **9.52** | **90.48** |

As can be seen form Table 5.3, there is a change from previous experiments and the best performance we achieved when Gaussian Mixture increased to 12 was 90.48% WER. Still, the performance we got was not very good.

In general, while we are using the language model which we have been built by HTK, our experiment was done using closed vocabulary. As a result, Table 5.4 shows both experiments conducted for context-independent (mono-phone based) and context-dependent (tri-phone based) 90.48% and 91.05% word error rate (WER), respectively.

Table 5.4 Result of Experiment using bigram language developed by HTK.

| Parameter tuning Values | | Number of Gaussian Mixture | Per cent of Word | | Word Error Rate (%WER) | Acoustic Models |
|---|---|---|---|---|---|---|
| $-p$ options | $-s$ options | | Correctly Recognized | Accurately Recognized | | |
| 1.0 | 15.0 | 12 | 11.90 | 9.52 | 90.48 | tri-phones |
| | | | 9.62 | 8.95 | 91.05 | mono-phones |

As can be seen from Table 5.4, the word error rate obtained were at high which shows the performance is very small.

By having the values of Gaussian Mixtures, the tuning parameters for word insertion penalty and grammar scale factor that we have the best performance in our previous experiments, we tried to conduct another experiment using the bigram language model developed by SRILM. Because SRILM was a well-known tool and has a good performance for language modeling, we have used this tool to build bigram language model. Additionally, we have conducted experiments for both context-dependent (tri-phones based) and context-independent (mono-phones based) acoustic models.

We have also used an open vocabulary, because Bisani and Ney, (2005) stated that a large vocabulary speech recognition systems operate with an open vocabulary. By doing so, Table 5.5 shows the result we obtained during the experiment.

Table 5.5 Result of Experiment using bigram language developed by SRILM

| Parameter tuning Values | | Number of Gaussian Mixture | Per cent of Word | | Word Error Rate (%WER) | Acoustic Models |
|---|---|---|---|---|---|---|
| $-p$ options | $-s$ options | | Correctly Recognized | Accurately Recognized | | |
| 1.0 | 15.0 | 12 | 12.05 | 10.05 | 89.95 | tri-phones |
| | | | 11.64 | 5.89 | 94.11 | mono-phones |

As shown in Table 5.5, the result we achieved has a change from prior experiments and still the performance was poor (89.95%WER). This is because the test data is taken from FBC, while the training mainly consists of data from other sources. Therefore, we have conducted another experiment by using test sentences prepared from all the sources of data used in our study. The proportion of test data when all sources used were presented in Table 5.6.

Table 5.6 Proportion of speech data used for training and testing

| Source | Length | Training | Testing |
|---|---|---|---|
| ORTO | 04:41:28 | 2003 | 225 |
| FBC | 00:40:56 | 275 | 25 |
| VOA | 00:53:14 | 375 | 50 |
| **Total** | **06:15:38** | **2653** | **300** |

Another experiment was conducting by taking the values of Gaussian numbers and options of -p and –s at that the highest performance was performed in prior experiments. Accordingly, the result of this experiment was presented in Table 5.7.

Table 5.7 Result of Experiment conducted using test data collected from all sources

| Parameter tuning Values | | Number of Gaussian Mixture | Per cent of Word | | Word Error Rate (%WER) | Acoustic Models |
|---|---|---|---|---|---|---|
| $-p$ options | $-s$ options | | Correctly Recognized | Accurately Recognized | | |
| 1.0 | 15.0 | 12 | 13.64 | 10.16 | 89.84 | tri-phones |
| | | | 12.32 | 8.54 | 91.46 | mono-phones |

From this result, we recognized as there is a change when test data were taken from all sources the training mainly involved. As can be seen from Table 5.7, we conclude that using test data from all sources that contained the trained data, using bigram language model that developed by SRILM was displays the best results from other experiments.

## 5.6. DISCUSSION OF RESULTS

Results that obtained from our findings were shown in earlier section. However, to discuss our result by comparing with other work, there is no similar work for the language (i.e., broadcast news speech recognition system was not conducted for Afaan Oromo before this study). But for Amharic language, Adugna (2015) developed spontaneous speech recognizer system using speech collected from web that recorded in local Medias and still it is not news and continuous speech.

Anyways, the result we obtained in this study was very poor when we compare with the earlier work. Moreover, this research was involves: a large vocabulary, continuous speech, and speaker-independent speech that make more difficult to develop the recognizer system from such type of speech. As we discussed in chapter 2 section 2, the performance of such type was less.

The other thing that make our result too small was; we have used an open vocabulary. This means the bigram language model we have built was not include the list of unique words that obtained from test transcriptions data.

## 5.7. CHALLENGES

Throughout the lifetime of this study, the researcher has faced with several challenges. The audio data collected from broadcast news was difficult by nature because they involve non-speech like background music and this is one of the challenges we have seen during the study. Additionally, because we have no any tool at hand that aids us to segment speech automatically, the researcher has segmented speech manually.

Several tools used in the work of Natural Language Processing (NLP) in general and speech recognition in specific were suggested with UNIX/LINUX operating system. Because of the machine that the researcher used stack while using this operating system, we were obliged to do the experiment on Windows operating system.

# CHAPTER SIX

# SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

## 6.1. INTRODUCTION

In this chapter we have put the summary of our study, conclusion of the study based on the research questions we have, and forwarded recommendations for further researchers.

## 6.2. SUMMARY

Investigating a Large Vocabulary, Speaker-independent, Continuous Speech Recognition (LVCSR) System for Afaan Oromo from Broadcast News speech corpus was the overall work of this study. In order to do this thesis work, we had consulted different literatures about features of the language, development of ASR from broadcast news in general and the related materials were also reviewed. For this study, from several ASR approaches, the widely used statistical (stochastic) approach and modeling technique the Hidden Markov Models (HMM) were used.

The corpus required for this study was collected from different sources. Accordingly, audio data from ORTO, VOA, and FBC; also text data from Bariisaa Afaan Oromo newspaper. The combination of this corpus helps us to show as our system could be work at different environment. This is because of the time constraints and scarcity of previous speech corpus for the language.

Audacity was used for audio preprocessing and segmenting the speech needed for the study. Also we used the HTK tools for extracting features of the speech, for training the system, for recognizing test utterances, and for analyzing the result of the recognizer. In order to develop the language model, SRILM language modeling tool was used and bigram language model was built. HLSTAT and HBUILD also used for building statistical bigram language model and constructing word network, respectively.

The speech recognizer system developed from 57 anchors or speakers (42 males and 15 females) using 2953 sentences which have 06:15:38 hours long. In addition text data about 2000 sentences used for language modeling purpose among collected from Bariisaa. Out of 2953 sentences, 2653 (10138 unique words dictionary) used for training

the speech recognizer system and remaining 300 sentence (2516 unique words dictionary) were used to test the developed system. Speakers who are involved in training does not involved in testing. Therefore, from 57 speakers only 12 speakers (9 males and 3 females) were participated for testing.

From several experiments done in this study, the best performance achieved was after increasing Gaussian Mixture to 12 and changing the options of $-p$ to 1.0 and$-s$ to 15.0, for word insertion penalty and grammar scale factor, respectively. The result shows that as context-dependent (tri-phone based) acoustic models perform the best results. As presented in Table 5.5, the word error rate obtained were 91.46% WER and 89.84% WER, for context-independent and context-dependent, respectively.

## 6.3. CONCLUSION

There are several challenges what we have faced in this study using broadcast news speech corpus. Preparing required corpus was one challenge. Because, the broadcast news speech was difficult by their nature (i.e., they include other non-speech like music background and additional telephone reports with interviews). Additionally, the decoder was not functional on the machine we have used for this study.

We have been prepared the speech corpus based on our scope and the resource we have for the study. By doing so we tried to overcome some challenges that happen in developing speech recognizer system for Afaan Oromo using broadcast news. Accordingly, non-speech that not required for our work was avoided during speech segmentation.

From obtained results, we have a high Word Error Rate (WER) of 91.46% and 89.84%, for context-independent and context-dependent, respectively. WER was high in this study, that shows as the accuracy of our recognizer system was very small (i.e., high word error rate shows low accuracy). However, we have gone to the most optimal level and very small improvements have seen. The researcher conducted several experiments in order to improve the performance of the recognizer; like increasing the Gaussian Mixtures, tuning parameters of word insertion penalty and grammar scale factors.

## 6.4. RECOMMENDATIONS

The researcher forwards the following recommendations for future work.

One of the limitations we have faced during this work was collecting speech corpus from different sources to prepare speech corpus for our study which take a lot of time. Accordingly, we have been prepared 5 and half hours for training and about 40 minutes of speech corpus for testing from broadcast news. But, literatures tells that using a large speech data was good to get better performance. Therefore, preparing such standardized speech corpus is recommended for further researches in the area of natural language processing particularly in speech recognition for Afaan Oromo.

The very expensive task in the work of speech recognition using broadcast news speech was speech segmentation, and we manually done it after spent a lot of time. So, developing automatic speech segmentation will be one issue for further researcher.

In this study, we have not tried to identify the variety of dialects. And also capitalization rule and punctuations were not used in both making transcription and text preparation (normalization) for language modeling. So, an interested researcher can conduct a research in the area by considering aforementioned limitations.

This study is the first of its kind to explore the possibility of developing continuous speech recognition system for Afaan Oromo using broadcast news speech corpus. However, the result obtained was yet a high WER. Therefore, we recommend to conduct a research on the area in order to increase the performance of ASR system using broadcast news speech corpus for Afaan Oromo.

We have been used a bi-gram language model. However, literatures reveals that using trigram language model was performs better than bigram language model. Although, because our decoder was not functionally work, we couldn't use trigram language model in the study. Thus, using trigram language model in order to conduct a research on the area can be a gap for future work.

From several approaches for natural language processing in general and for speech recognition in particular; due to time limitation, only statistical or stochastic approach was experimented with HMM modeling technique. We therefore recommend that conducting the experiment with other approaches like Artificial Neural Networks, Acoustic-phonetic and hybrid of these approaches using the prepared speech corpus.

# REFERENCES


Abdella Kemal. (2010). *Speaker Dependent Speech Recognition for Sidaama Language.* Master's thesis, Addis Ababa University, Addis Ababa, Ethiopia.

Abraham Tesso. (2014). *Challenges of Diacritical Marker or Hudhaa Character in Tokenization of Oromo Text.* Journal of Software, 9(7).

Addunyaa Barkeess. (2012). *NATOO: Yaadrimee Caasluga Afaan Oromoo (Concept of Afaan Oromo Grammar).* Finfinnee, Oromiyaa.

Addunyaa Barkeessaa. (2014). *Seemmoo: Bu'uura Barnoota Afaanii fi Afoola Oromoo*. Oromiyaa, Finfinnee, Far East Trading P.L.C

Adugna Deksiso. (2015). *Spontaneous Speech Recognition for Amharic Using HMM.* Master's thesis, Addis Ababa University, Addis Ababa, Ethiopia.

Alexander, S. (2003). *Efficient Methods for Automatic Speech Recognition.* Doctoral Dissertation, Royal Institute of Technology Stockholm, Stockholm, Sweden.

Ashenafi Demissie. (2009). *A Speech Recognition System for Afaan Oromoo.*Master's thesis, Addis Ababa University, Addis Ababa, Ethiopia.

Bisani, M. and Ney, H. (2005). Open vocabulary speech recognition with flat hybrid models, in Inter-speech, Lisbon, Portugal. pp. 725 – 728.

Daniel, H. (2008). *Audio Pre-processing and Speech Recognition for Broadcast News,* UniversidadeT´Ecnica De LisboaInstituto Superior T´Ecnico

FDRE Population census Commission, Summary and Statistical report of the 2007 population and housing census. Printed by United Nations Population Fund (UNFPA) Addis Ababa, Ethiopia.

Hafte Abera. (2009). *Hidden Markov Model Based Tigrigna Speech Recognition*, Master's thesis, Addis Ababa University, Addis Ababa, Ethiopia.

Hank, H. (2013). *Introduction to SRILM Toolkit. Department of Computer Science & Information Engineering.* National Taiwan Normal University.

Hinsene Makuria. (2009).*Elellee Conversation: Afaan Oromo writing system.* Addis Abba, Ethiopia, Commercial printing E.pp 23-39.

Jurafsky, D., & Martin, J.H. (2007). *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice-Hall,Inc.

Jurafsky, D., & Martin, J.H. (2014). *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice-Hall, Inc.

Juraj, K. (2004). White paper "HTK vs. SPHINX for SPEECH Recognition" by Ilkovičová 3, Bratislava, Slovakia (visited at: 19/02/2015)

Kassahun Gelana. (2010) *A Continuous, Speaker Independent Speech Recognizer for Afaan Oromoo*.Master's thesis, Addis Ababa University, Addis Ababa, Ethiopia.

Mirressa Amanu. (2014). *Qalbii: Seerluga Oromoo.* Finfinnee, Oromiyaa, ELLENI P.P.PLC, PP. 32 – 41.

Murat, A. (2003). A *Large Vocabulary Continuous Speech Recognition for Turkish Using HTK,* Master's thesis, the Middle East Technical University.

Rabiner, L.R. (1989). *A tutorial on Hidden Markov Model and selected applications in speech recognition*, Preceding of IEEE. 77(2).

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

Rabiner, L.R., and Juang, B.H. (2004). *Automatic Speech Recognition, A Brief History of the Technology Development*. Accessed on 05/01/2016, 04:32:19 from http://www.idi.ntnu.no/~gamback/teaching/TDT4275/literature/juang_rabiner04.pdf

Solomon Teferra. (2005). *Automatic Speech Recognition for Amharic*. Doctoral Dissertation, Hamburg University, Germany.

Source for Oromo alphabets IPA accessed on 20/06/2016 from *www.sas.upenn.edu/African_Studies/Hornet/Afaan_Oromo_19777.html*

Suman, K. et. al. (2015*).A Review on Different Approaches for Speech Recognition System.* International Journal of Computer Applications, Volume 115(22).

Teferi Degeneh (2015). *The Development of Oromo Writing System*. Doctoral Dissertation, School of European Culture and Languages, University of Kent.

Teferi Kebebew. (2010). *Speech Recognition for Afan Oromo Using Hybrid Hidden Markov Models and Artificial Neural Networks.* Master's thesis, Addis Ababa University, Addis Ababa, Ethiopia.

Tilahun Gamta. (1993). *Qubee Afaan Oromoo: Reasons for Choosing the Latin Script for Developing an Oromo Alphabet.*The Journal of OromoStudies, 1(1) pp: 36.

Tilahun Gamta. (2004). *Pluralization in Afaan Oromoo.* The Journal of Oromo Studies, 11(1 and 2) pp: 29 - 45.

Young, Steve et. al. (2006). *The HTK Book for HTK Version 3.4.1,* Cambridge University Engineering Department.

Zegaye Seifu. (2003). *Hidden Morkov Model Based Large Vocabulary, Speaker Independent, continuous Amharic Speech Recognition.* Master's thesis, Addis Ababa University, Addis Ababa, Ethiopia.

# APPENDICES

## Appendix A: Summary of Tools Used in the Study

| Tools or Software | Version | Purpose |
|---|---|---|
| HTK | 3.4.1 | To implement the speech recognizer |
| Visual Studio | 12 | HTK needs C:\Program Files \Microsoft Visual Studio\VC\bin \ for installation |
| Active Perl | 5.20.2 | For testing the installed HTK-3.4.1 |
| 7-zip | 16.02 | For unpacking the HTK source code archive |
| Notepad++ | 6.9.1.0 | To write transcription and pronunciation |
| Audacity | 2.1.2 | For Audio Pre-processing |
| Command Prompt | | To run/execute HTK command |
| Microsoft Word | 10 | To write the document/report of the study |
| Microsoft PowerPoint | 10 | For making presentation |
| SRILM | 1.7.1 | For developing language model |

## Appendix B: Samples of prompts trainprompt.txt and testprompt.txt files

**trainprompt.txt file**

*/train1 yuuniyeeniin kun wayita ammaa kana garuu tajaajila isaa baballiseera

*/train2 isaan keessaa galii maatii qonnaan bultootaa dabaluufis horsiisa lukkuu irratti xiyyeeffatee hojjetaa jiraa

*/train3 bara kanas dabalataan caaccuu kuma sadii ol kunuunsuun baatii lama yoo gahan keessattuu dubartoota miseensa yuuniyeenichaa tahhaniif gati madaalawaan dhiyeessuun dubartoonni dinagdee isaaniitiin akka cimanii fi jedhan

*/train4 obbo hayiluu xilahuun hojjetaa giddu gala kanaati

*/train5 kun immoo keessumaa dubartoota dinagdeen cimsuuf gargaara

…

**testprompt.txt file**

*/test1 fooramiin biizinasii itiyookeeniyaa naayiroobiitti gaggeeffamaa jiraa

*/test2 marii kana irratti muummichi ministeraa hayila maaram dassaalenyii fi perezedaantiin keeniyaa uhuruu keeniyaattaan argamanii jiruu

*/test3 itti aanaan perezedaantii keeniyaa wiiliyaam ruuttoo fi ministirri dhimma alaa dokteer tewoodiroos adahaanoom dabalatee miseensonni hawwaasaa daldalaa keeniyaatti mariyataa jiruu

*/test4 dura taahhaan waldaa indaastirii damee oomishaa keeniyaa filooraa muutaan akka jedhanitti abbootiin qabeenyaa keeniyaa itoophiyaa keessatti hojiilee inveestimentii irratti bobbahuu ni barbaaduu

*/test5 daldala biyyoota lameenii cimsuufis mootummoonni biyyoota lamaanii ijaarsa buhhuuraalee misoomaa irrattis cimanii akka hojjetan gaafataniiru jechuun tajaajilli oduu itoophiyaa gabaaseera

…

## Appendix C: Sample of Pronunciation Dictionary

```
aadaa [aadaa] aa d aa
afuura [afuura] a f uu r a
afuura [hafuura] h a f uu r a
ammo [ammo] a mm oo
ammo [immo] i mm oo
arfan [arfan] a r f a n
arfan [afran] a f r a n
itiyoophiyaa [itiyoophiyaa] i t i y oo ph i y aa
itiyoophiyaa [itoophiyaa] i t oo ph i y aa
itiyoophiyaa [toophiyaa] t oo ph i y aa
itiyoophiyaa [xoophiyaa] x oo ph i y aa
miliyoona [miliyoona] m i l i y oo n a
miliyoona [miliyeena] m i l i y ee n a
miliyoona [miiliyoona] m ii l i y oo n a
miliyoona [miiliyeena] m ii l i y ee n a
salgan [salgan] s a l g a n
salgan [saglan] s a g l a n
saglaffaa [saglaffaa] s a g l ff aa
saglaffaa [salgaffaa] s a l g ff aa
tokko [tokko] t o kk o
tokko [takka] t a kk a
qofaa [qofaa] q o f aa
qofaa [qobaa] q o b aa
qofaa [kophaa] k o ph aa
uumaa [uumaa] uu m aa
uumaa [huumaa] h uu m aa
```

## Appendix D: Number of Alternative pronunciation dictionary

| NO | Word | NO. of pronunciation |
|---|---|---|
| 1 | **Aarii** | 2 |
| 2 | **Abaaboo** | 2 |
| 3 | **Abidda** | 2 |
| 4 | **Afuura** | 2 |
| 5 | **Amboo** | 2 |
| 6 | **Ammoo** | 2 |
| 7 | **Arfaffaa** | 3 |
| 8 | **Ariyuu** | 2 |
| 9 | **Awwaasaa** | 2 |
| 10 | **Bahe** | 2 |
| 11 | **Barrulee** | 2 |
| 12 | **Biliyoona** | 4 |
| 13 | **Bishaan** | 2 |
| 14 | **Boru** | 3 |
| 15 | **Dhibba** | 2 |
| 16 | **Digdama** | 2 |
| 17 | **Eeyyema** | 2 |
| 18 | **Gibtsi** | 2 |
| 19 | **Gurguddaa** | 2 |
| 20 | **Haga** | 2 |
| 21 | **Hamma** | 2 |
| 22 | **Hawwaasa** | 2 |
| 23 | **Ishee** | 3 |
| 24 | **Itiyoophiyaa** | 4 |
| 25 | **Jaha** | 3 |
| 26 | **Kaleessa** | 2 |
| 27 | **Kana** | 2 |
| 28 | **Keenya** | 2 |
| 29 | **Kompiitera** | 2 |
| 30 | **Kun** | 2 |
| 31 | **Kurnaffaa** | 2 |
| 32 | **Miliyoona** | 4 |
| 33 | **Mokonnon** | 2 |
| 34 | **Omisha** | 2 |
| 35 | **Otoo** | 3 |
| 36 | **Qofaa** | 3 |
| 37 | **Saglaffaa** | 2 |
| 38 | **Salgan** | 2 |
| 39 | **Tahuu** | 2 |
| 40 | **Tokko** | 2 |
| 41 | **Uumaa** | 2 |
| 42 | **Waajira** | 2 |
| 43 | **Yennee** | 2 |
| 44 | **Yommuu** | 2 |
| 45 | **Yuunivarsiitii** | 5 |

## Appendix E: Sample of coding the data for traincode.scp and testcode.scp

train/audio/onew1.wav train/mfcc/onew1.mfc

train/audio/onew2.wav train/mfcc/onew2.mfc

train/audio/onew3.wav train/mfcc/onew3.mfc

train/audio/onew4.wav train/mfcc/onew4.mfc

train/audio/onew5.wav train/mfcc/onew5.mfc

train/audio/onew6.wav train/mfcc/onew6.mfc

train/audio/onew7.wav train/mfcc/onew7.mfc

train/audio/onew8.wav train/mfcc/onew8.mfc

train/audio/onew9.wav train/mfcc/onew9.mfc

…….

test/audio/test1.wav test/mfcc/test1.mfc

test/audio/test2.wav test/mfcc/test2.mfc

test/audio/test3.wav test/mfcc/test3.mfc

test/audio/test4.wav test/mfcc/test4.mfc

test/audio/test5.wav test/mfcc/test5.mfc

test/audio/test6.wav test/mfcc/test6.mfc

test/audio/test7.wav test/mfcc/test7.mfc

test/audio/test8.wav test/mfcc/test8.mfc

test/audio/test9.wav test/mfcc/test9.mfc

…….

## Appendix F: Different configuration tools

**config_hcopy**

```
TARGETKIND = MFCC_0_D_A
SOURCEFORMAT = WAV
TARGETFORMAT = HTK
SOURCERATE=625
TARGETRATE = 100000.0
SAVECOMPRESSED = TRUE
SAVEWITHCRC = TRUE
WINDOWSIZE = 250000.0
USEHAMMING = TRUE
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = FALSE
```

**config_hcompv**

```
TARGETKIND = MFCC_0_D_A
SOURCEFORMAT = HTK
SOURCERATE = 625
TARGETRATE = 100000.0
SAVECOMPRESSED = TRUE
SAVEWITHCRC = TRUE
WINDOWSIZE = 250000.0
USEHAMMING = TRUE
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = FALSE
```

**config_hvite**

```
TARGETKIND = MFCC_0_D_A
TARGETRATE = 100000.0
SAVECOMPRESSED= TRUE
SAVEWITHCRC = TRUE
WINDOWSIZE = 250000.0
USEHAMMING = TRUE
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = FALSE
SOURCEFORMAT = HTK
USESILDET = TRUE
MEASURESIL = FALSE
OUTSILWARN = TRUE
MICIN= TRUE
```

## Appendix G: Prototype of proto files

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"
<BeginHMM>
<NumStates> 5
<State> 2
<Mean> 39
      0.0 0.0 0.0 0.0 0.0…
<Variance> 39
      1.0 1.0 1.0 …
<State> 3
<Mean> 39
      0.0 0.0 0.0 …
<Variance> 39
      1.0 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 …
<Variance> 39
      1.0 0 1.0 1.0 1.0 1.0 …
<TransP> 5
  0.0 1.0 0.0 0.0 0.0
  0.0 0.6 0.4 0.0 0.0
  0.0 0.0 0.6 0.4 0.0
  0.0 0.0 0.0 0.7 0.3
  0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

## Appendix H: Sample of the tree.hed edit script

```
RO 100 "stats"

TR 0

QS  "R_NonBoundary"     { *+* }

QS  "R_Silence"         { *+sil }

QS  "R_Stop" { *+p,*+ph,*+d,*+b,*+t,*+j,*+dh,*+k,*+q,*+g,*+ch,*+x,*+c, }

QS  "R_Nasal"           { *+m,*+mm,*+n,*+nn,*+ny }

QS  "R_Fricative"       { *+s,*+ss,*+sh,*+z,*+f,*+ff,*+hh,*+v }

QS  "R_Vowel"           { *+a,*+aa,*+e,*+ee,*+i,*+ii,*+o,*+oo,*+u,*+uu }

QS  "R_C-Front"           { *+p,*+ph,*+b,*+m,*+*+f,*+v,*+w }

QS  "R_C-Central"     { *+t,*+dd,*+d,*+dh,*+x,*+n,*+s,*+z,*+r }

QS  "R_C-Back"            { *+sh,*+ch,*+j,*+jj,*+y,*+k,*+kk,*+g,*+ny,hh }

QS  "R_V-Front"         { *+i,*+ii,*+e,*+ee }

QS  "R_V-Central"       { *+a }

QS  "R_V-Back"          { *+u,*+aa,*+oo,*+uu,*+o }

QS  "R_Unvoiced-Cons"   { *+p,*+tt,*+k,*+kk,*+ch,*+f,*+ff,*+s,*+ss,*+sh }

QS  "R_Voiced-Cons"     { *+j,*+b,*+bb,*+d,*+dd,*+dh,*+g,*+gg,*+v,*+z }

QS  "R_Long"            { *+aa,*+ee,*+ii,*+oo,*+uu }

QS  "R_Short"           { *+a,*+e,*+i,*+o,*+u }

QS  "R_IVowel"          { *+i,*+ii }

QS  "R_EVowel"          { *+e,*+ee }

QS  "R_AVowel"          { *+a,*+aa }

QS  "R_OVowel"          { *+o,*+oo, }

QS  "R_UVowel"          { *+u,*+uu }

QS  "R_Voiced-Stop"     { *+b,*+bb,*+d,*+dd,*+g,*+gg,*+j,*+jj }

QS  "R_Unvoiced-Stop"   { *+p,*+pp,*+t,*+tt,*+k,*+kk,*+ch }

QS  "R_Voiced-Fric"     { *+z,*+v }

QS  "R_Unvoiced-Fric"   { *+s,*+sh,*+th,*+f,*+ch }

QS  "R_Front-Fric"      { *+f,*+ff,*+v }
```

```
QS  "R_Central-Fric"    { *+s,*+ss,*+z }

QS  "R_Back-Fric"       { *+sh,*+ch }

QS "R_a"          { *+a }               QS "R_gg"         { *+gg }

QS "R_aa"         { *+aa }              QS "R_h"          { *+h }

QS "R_b"          { *+b }               QS "R_hh"         { *+hh }

QS "R_bb"         { *+bb }              QS "R_i"          { *+i }

QS "R_c"          { *+c }               QS "R_ii"         { *+ii }

QS "R_cc"         { *+cc }              QS "R_j"          { *+j }

QS "R_ch"         { *+ch }              QS "R_jj"         { *+jj }

QS "R_d"          { *+d }               QS "R_k"          { *+k }

QS "R_dd"         { *+dd }              QS "R_kk"         { *+kk }

QS "R_dh"         { *+dh }              QS "R_l"          { *+l }

QS "R_e"          { *+e }               QS "R_ll"         { *+ll }

QS "R_ee"         { *+ee }              QS "R_m"          { *+m }

QS "R_f"          { *+f }               QS "R_mm"         { *+mm }

QS "R_ff"         { *+ff }              QS "R_n"          { *+n }

QS "R_g"          { *+g }               QS "R_nn"         { *+nn }

… … … … … …  … … … … … … … … … … … … … … … … … … …  … … … …

… … … … … …  … … … … … … … … … … … … … … … … … … …  … … … …

… … … … … …  … … … … … … … … … … … … … … … … … … …  … … … …

QS  "L_NonBoundary"     { *-* }

QS  "L_Silence"         { sil-* }

QS  "L_Stop" { p-*,ph-*,d-*,b-*,t-*,j-*,dh-*,k-*,q-*,g-*,ch-*,x-*,c-* }

QS  "L_Nasal"           { m-*,mm-*,n-*,nn-*,ny-* }

QS  "L_Fricative"       { s-*,ss-*,sh-*,z-*,f-*,ff-*,hh-*,v-* }

QS  "L_Vowel"           { a-*,aa-*,e-*,ee-*,i-*,ii-*,o-*,oo-*,u-*,uu-* }

QS  "L_C-Front"           { p-*,ph-*,b-*,m-*,f-*,v-*,w-* }

QS  "L_C-Central"     { t-*,dd-*,d-*,dh-*,x-*,n-*,s-*,z-*,r-* }

QS  "L_C-Back"            { sh-*,ch-*,j-*,jj-*,y-*,k-*,kk-*,g-*,ny-*,hh-* }

QS  "L_V-Front"         { i-*,ii-*,e-*,ee-* }
```

```
QS  "L_V-Central"       { a-* }
QS  "L_V-Back"          { u-*,aa-*,oo-*,uu-*,o-* }
QS  "L_Unvoiced-Cons"   { p-*,t-*,k-*,kk-*,ch-*,f-*,ff-*,s-*,ss-*,sh-* }
QS  "L_Voiced-Cons"     { j-*,b-*,bb-*,d-*,dd-*,dh-*,g-*,gg-*,v-*,z-* }
QS  "L_Long"            { aa-*,ee-*,ii-*,oo-*,uu-* }
QS  "L_Short"           { a-*,e-*,i-*,o-*,u-* }
QS  "L_IVowel"          { i-*,ii-* }
QS  "L_EVowel"          { e-*,ee-* }
QS  "L_AVowel"          { a-*,aa-* }
QS  "L_OVowel"          { o-*,oo-* }
QS  "L_UVowel"          { u-*,uu-* }
QS  "L_Voiced-Stop"     { b-*,bb-*,d-*,dd-*,g-*,gg-*,j-*,jj-* }
QS  "L_Unvoiced-Stop"   { p-*,pp-*,t-*,tt-*,k-*,kk-*,ch-* }
QS  "L_Voiced-Fric"     { z-*,v-* }
QS  "L_Unvoiced-Fric"   { s-*,sh-*,th-*,f-*,ch-* }
QS  "L_Front-Fric"      { f-*,ff-*,v-* }
QS  "L_Central-Fric"    { s-*,ss-*,z-* }
QS  "L_Back-Fric"       { sh-*,ch-* }
QS  "L_a"           { a-* }             QS  "L_f"          { f-* }
QS  "L_aa"          { aa-* }            QS  "L_ff"         { ff-* }
QS  "L_b"           { b-* }             QS  "L_g"          { g-* }
QS  "L_bb"          { bb-* }            QS  "L_gg"         { gg-* }
QS  "L_c"           { c-* }             QS  "L_h"          { h-* }
QS  "L_cc"         { cc-* }             QS  "L_hh"         { hh-* }
QS  "L_ch"         { ch-* }             QS  "L_i"          { i-* }
QS  "L_d"          { d-* }              QS  "L_ii"         { ii-* }
QS  "L_dd"         { dd-* }             QS  "L_j"          { j-* }
QS  "L_dh"         { dh-* }             QS  "L_jj"         { jj-* }
QS  "L_e"          { e-* }              QS  "L_k"          { k-* }
QS  "L_ee"         { ee-* }             QS  "L_kk"         { kk-* }
```

```
… … … … … …  … … … … … … … … … … … … … … … … … …  … … … …
… … … … … …  … … … … … … … … … … … … … … … … … …  … … … …
… … … … … …  … … … … … … … … … … … … … … … … … …  … … … …
TR 2
TB 350.0 "ST_aa_2_" {("aa","*-aa+*","aa+*","*-aa").state[2]}
TB 350.0 "ST_d_2_" {("d","*-d+*","d+*","*-d").state[2]}
TB 350.0 "ST_sp_2_" {("sp","*-sp+*","sp+*","*-sp").state[2]}
… … … … … …  … … … … … … … … … … … … … … … … … …  … … … …
TB 350.0 "ST_ph_2_" {("ph","*-ph+*","ph+*","*-ph").state[2]}
TB 350.0 "ST_sil_2_" {("sil","*-sil+*","sil+*","*-sil").state[2]}
TB 350.0 "ST_aa_3_" {("aa","*-aa+*","aa+*","*-aa").state[3]}
TB 350.0 "ST_d_3_" {("d","*-d+*","d+*","*-d").state[3]}
TB 350.0 "ST_sp_3_" {("sp","*-sp+*","sp+*","*-sp").state[3]}
… … … … … …  … … … … … … … … … … … … … … … … … …  … … … …
… … … … … …  … … … … … … … … … … … … … … … … … …  … … … …
… … … … … …  … … … … … … … … … … … … … … … … … …  … … … …
TB 350.0 "ST_ph_3_" {("ph","*-ph+*","ph+*","*-ph").state[3]}
TB 350.0 "ST_sil_3_" {("sil","*-sil+*","sil+*","*-sil").state[3]}
TB 350.0 "ST_aa_4_" {("aa","*-aa+*","aa+*","*-aa").state[4]}
TB 350.0 "ST_d_4_" {("d","*-d+*","d+*","*-d").state[4]}
TB 350.0 "ST_sp_4_" {("sp","*-sp+*","sp+*","*-sp").state[4]}
… … … … … …  … … … … … … … … … … … … … … … … … …  … … … …
… … … … … …  … … … … … … … … … … … … … … … … … …  … … … …
… … … … … …  … … … … … … … … … … … … … … … … … …  … … … …
TB 350.0 "ST_ph_4_" {("ph","*-ph+*","ph+*","*-ph").state[4]}
TB 350.0 "ST_sil_4_" {("sil","*-sil+*","sil+*","*-sil").state[4]}
TR 1
AU "fulllist"
CO "tiedlist"
ST "trees"
```