



ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

**Large Vocabulary Continuous Speech Recognition System for
Afaan Oromo using Hidden Markov Model (HMM)**

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN
INFORMATION SYSTEMS

BY: Duresa Deksiso Geleto

September, 2016

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

**Large Vocabulary Continuous Speech Recognition System for
Afaan Oromo using Hidden Markov Model (HMM)**

BY: Duresa Deksiso Geleto

Signature of the board of the examiners for approval

NAME

SIGNATURE

DATE

ACKNOWLEDGMENT

First of all, I would like to thank God for supporting and being with me in all walks of my life.

I would like to thank my advisor Dr. Solomon Teferra for his constructive comments, guidance leadership and support. He has encouraged me to work with confidence and learn how to solve problems. He was patient with me and worked with commitment during the whole life of this research.

My special thanks go to my wife Adde Hawine Shila, My brother Adugna Deksiso and my friend Bantegize Addis for their invaluable contribution during the course of the study.

I also would like to thank Goro secondary and preparatory school students and staff member for their volunteer to read speech corpus I used in this study.

Finally, my thanks also goes to my friends who have contributed for this research more than just being a friend.

Table of Contents

ACKNOWLEDGMENT.....	II
Table of Contents.....	III
List of Tables	VI
List of Figure.....	VII
Acronyms.....	VIII
ABSTRACT.....	IX
CHAPTER ONE	1
INTRODUCTION.....	1
1.1 General Background.....	1
1.2 Classification of Automatic speech recognition.....	4
1.3 Approaches in Automatic Speech Recognitions	6
1.4 Statement of the Problem and Its Justification.....	8
1.5 Research Questions	9
1.6 Objective of the Study.....	9
1.6.1 General Objective	9
1.6.2 Specific Objectives	9
1.7 Scope and Limitations of the Study	10
1.8 Methodology of the Study.....	11
1.9 Significance of the study	13
1.10 Organization of the Thesis	14
CHAPTER TWO	15
LITERATURE REVIEW	15
2.1 Basics of Speech Recognition.....	15
2.2 Overview of Automatic Speech Recognition.....	16
2.3 Speech Recognition Tasks	18
2.3.1 Preprocessing	18
2.3.2 Recognition	19
2.4 Automatic Speech Recognition Types	20

2.5	Signal Processing and Feature Extraction.....	21
2.5.1	Signal Processing.....	21
2.5.2	Feature Extraction.....	22
2.5.3	Production Based Analysis	22
2.5.4	Perception Based Analysis.....	23
2.6	Acoustic Models.....	24
2.6.1	Hidden Markov Model (HMM).....	25
2.7	Lexical Models.....	31
2.8	Language Models	32
2.9	Speech Units in Automatic Speech Recognition.....	33
2.10	Related works.....	35
2.10.1	Review of Works done so far on Amharic ASR.....	35
2.10.2	Review of Works done so far on Afaan Oromo ASR.....	37
CHAPTER 3		39
THE AFAAN OROMO LANGUAGE		39
3.1	The Overview of Afaan Oromo Language.....	39
3.2	Afaan Oromo Alphabets and Orthography	40
3.3	Basics of Afaan Oromo Phonetics	42
3.4	Afaan Oromo Writing System.....	47
3.4.1	Spelling Rules	48
3.4.2	Grammar	49
CHAPTER 4		55
CMU SPHINX: THE RECOGNIZER LIBRARY		55
4.1	Why Sphinx.....	55
4.2	Acoustic Modelling Toolkit	56
4.3	Language Modelling Tool kit.....	56
4.4	Decoding Toolkit.....	57
CHAPTER 5		63
EXPERIMENTATION AND ANALYSIS		63
5.1	System Architecture	63

5.2	Data Preparation	64
5.3	Pronunciation Dictionary	65
5.4	Acoustic Model	66
5.4.1	Signal Segmentation and Annotation.....	66
5.4.2	Feature Extraction.....	67
5.4.3	Acoustic Model Training Files	68
5.5	Language Model.....	71
5.6	Recognizer Testing and Evaluation.....	72
5.7	Challenges	80
CHAPTER 6		81
CONCLUSION AND RECOMMENDATION		81
6.1	Conclusion.....	81
6.2	Recommendation.....	83
REFERENCES		84
Appendix A: Some Afaan Oromo Text Corpus.....		88
Appendix B: Afaan Oromo Canonical Pronunciation Dictionary		89
Appendix C: Afaan Oromo Phone Set.....		90
Appendix D: Training Configuration.....		90

List of Tables

Table 3. 1 Afaan Oromo Alphabets	41
Table 3. 2 Plosive Phonemes with Examples	43
Table 3. 3 Fricative Phonemes with Examples	44
Table 3. 4 Affricate Phonemes with Examples.....	44
Table 3. 5 Nasal Phonemes with Examples	44
Table 3. 6 Liquid Phonemes with Examples	45
Table 3. 7 Glide Phonemes with Examples	45
Table 3. 8 Loan Phonemes.....	45
Table 3. 9 Afaan Oromo Consonants (Extracted From: IPA)	46
Table 3. 10 Afaan Oromo Vowels	47
Table 3. 11 Jabaa and Laafaa, and Gabaabaa and Dheeraa	48
Table 5. 1 Parameters Used in Feature Extraction.....	68
Table 5. 2 Speaker Dependent Using 8 Gaussian Mixture with Skip.....	74
Table 5. 3 Speaker Dependent Using 8 Gaussian Mixture with Skip.....	74
Table 5. 4 Speaker Independent Using 8 Gaussian Mixture with Skip	75
Table 5. 5 Speaker Independent Using 8 Gaussian Mixture with Skip	75
Table 5. 6 Speaker Dependent Using 8 Gaussian Mixture without Skip.....	76
Table 5. 7 Speaker Dependent Using 8 Gaussian Mixture without Skip.....	77
Table 5. 8 Speaker Independent Using 8 Gaussian Mixture without Skip	77
Table 5. 9 Speaker Independent Using 8 Gaussian Mixture without Skip	78

List of Figure

Figure 1. 1 The Main Components of an Automatic Speech Recognition System [8].....	4
Figure 2. 1 Architecture of an ASR system based on statistical approach, adapted from [13]	17
Figure 4 1 Sphinx-4 Decoder Framework. The main blocks are the Front End, the Decoder, and the Linguist. Adapted from [39]	58
Figure 4 2 Sphinx-4 Front End. The Front End comprises one or more parallel chains of communicating Data Processors Adapted from [39]	58
Figure 5. 1 Afaan Oromo ASR System Architecture Using CMU Sphinx	63
Figure 5. 2 Praat Speech Segmentation, Labeling, and Transcription Software	67

Acronyms

ANN	Artificial Neural Network
AO	Afaan Oromo
AOASR	Afaan Oromo Automatic Speech Recognition
ASR	Automatic Speech Recognition
CV	Consonant Vowel
HSR	Human Speech Recognition
HMM	Hidden Markov Model
HTK	Hidden Markov Toolkit
LVCSR	Large Vocabulary Continuous Speech Recognition
MFCC	Mel-frequency Cepstrum Coefficients
NLP	Natural language processing
SOV	Subject Object Verb
WER	Word Error Rate
SER	Sentence Error Rate

ABSTRACT

The ultimate goal of automatic speech recognition is towards developing a model that automatically converts speech utterance into a sequence of words. Having similar objective of transforming Afaan Oromo language speech into its equivalent sequence of words, this study explored the possibility of developing large vocabulary continuous read speech recognizer prototype for Afaan Oromo using Hidden Markov Model.

Afaan Oromo is an Afro-Asiatic language, and the most widely spoken of the Cushitic family. It is spoken as a first language by more than 40 million Oromo and neighboring peoples in Ethiopia and Kenya. Afaan Oromo uses Latin script or characters (Roman orthography). The script of Afaan Oromo is called “**QUBEE**”. Afaan Oromo is a phonetic language, which means that it is spoken in the way it is written.

To develop and test the required speech recognizer performance one hour read speech data was collected, it was segmented and labeled into sentences with freely available toolkits. Prototype of large vocabulary continuous speech recognition system for Afaan Oromo were modeled using CMU sphinx open source speech recognition toolkit.

We have tested the recognizer by using two types of test data set. The first is speaker dependent test data set in which speaker that participated in training were participated in testing. The second is speaker independent test data set in which speaker participated in testing were do not participated in training.

The experimentation were done in two distinct phases. The first phase experimentation were done using with skip transition topology, context dependent tri-phone and using 8 Gaussian mixture. The second phase experimentation were done using without skip transition topology, context dependent tri-phone and using 8 Gaussian mixture. The language model we have used for this study were both bigram and tri-gram.

From the experimentation result bigram language model performed the highest word accuracy 93% for speaker dependent test data set and 43.6% word accuracy for speaker independent test data set.

CHAPTER ONE

INTRODUCTION

1.1 General Background

Speech is the most natural means of communication between humans it can be done without any tools or any explicit education. From previous several decades human beings tried to create technologies that could recognize correct speech. While humans can differentiate speech very easily, they in fact make use of much acoustic, linguistic and contextual information. It has been seen that relation between physical speech signal and the corresponding words is so much complex and very hard to understand. Both the research areas of automatic speech recognition (ASR) and human speech recognition (HSR) observe the recognition process from the acoustic signal to a series of recognized units. For ASR, the objective is to automatically transcribe the speech signal in terms of a sequence of items as close as possible [1].

Speech and hearing have involved as a main tool of communication among human beings [2]. The basic building block of the speech of any language is a set of sounds called phonemes. Starting from our childhood we learn the skill of communication naturally till we grow up which is very complicated. Even with the difference in terms of accent, articulation, nasality, roughness, volume, pitch, pronunciation, and speed, we are still able to interpret the speech most of the time as long as the spoken language is the language that we are familiar with [2].

Due to the familiarity to spoken language, we would also hope to interact with machines via speech. Scientists and researchers are finding their ways to produce an efficient speech recognizer so that a natural human-machine interface could be invented that replace the primitive interfaces, such as keyboard and mouse for the computer. With the existence of this human-machine interface, valuable applications would come into our life to make jobs done easier and effectively.

Speech Recognition also known as Automatic speech recognition (ASR) is defined as a process of converting a speech signal into a set of words by a certain algorithm that can be implemented as a

system program or a process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words [3].

Speech is the vocalized form of human communication. To communicate with a machine we still require interfaces like keyboard, mouse and screen etc., operated with the help of software. A simple alternative to these hardware interfaces is a software interface i.e. an ASR (Automatic Speech Recognition) system [1]. Automatic speech recognition is the task of taking an utterance of speech signal as an input, captured by a microphone, a telephone etc., and convert it into a text sequence as close as possible to the spoken data [4]. The main difficulties in implementation of an ASR system are due to different speaking styles of human beings and environmental disturbances. So the main aim of an ASR system is to transform a speech signal into text message independent of the device, speaker or the surroundings in an accurate and efficient manner.

Speech is the primary communication medium between people. This communication process has a complex structure consisting not only of the transmission of voice. Gestures, the language, the subject and the capability of the listener contribute to this process. As the maxim says, what you can tell is restricted to what the auditor can understand. In this respect, the performance of a speech recognizer system heavily depends on how and for which task you designed it.

Speech recognition area of science has its roots in the idea of communicating with a machine by voice. Our ability to communicate with machines and computers through keyboards or other devices is slower and more cumbersome. Speech can be regarded as an important component in order to make this communication easier.

Actually, a computer or a machine is not expected to understand what is uttered. But it is expected to be controlled via speech or to transcript the acoustic signal to symbols. The ultimate goal of research on ASR is to build machines that are indistinguishable from humans in the ability to communicate in natural spoken language. In this sense, speech recognition is not a mature science but an emerging one.

The earliest attempts to build ASR systems were made in the 1950's, when researchers tried to exploit the fundamental ideas of acoustic-phonetics. In 1952, a system for isolated digit recognition for a single speaker was built [1].

The early ASR devices applied threshold logic to voltage analogs of acoustic patterns to recognize words or short utterances of a certain speaker. It was understood then that no simple generalization of this technique would be sufficient to recognize the fluent speech. Indeed, it was well understood that the intelligence borne by the speech signal was encoded in a highly complex way [5].

Speech technology is the technology of today and tomorrow with a developing number of methods and tools for better implementation. Speech recognition has a number of practical implementations for both fun and serious works. Automatic speech recognition has an interesting and useful implementation in expert systems, a technology whereby computers can act as a substitute for a human expert. An intelligent computer that acts, responds or thinks like a human being can be equipped with an automatic speech recognition module that enables it to process spoken information. Medical diagnostic systems, for example, can diagnose a patient by asking him a set of questions, the patient responding with answers, and the system responds with what might be a possible disease [6].

Figure 1.1 illustrates the major modules of an ASR system and their relation to applications. In feature extraction, signal processing techniques are applied to the speech signal in order to dig out the features that distinguish different phonemes from each other. Given the features extracted from the speech, acoustic modeling provides probabilities for different phonemes at different time instants. Language modeling, on the other hand, defines what kind of phoneme and word sequences are possible in the target language or application at hand, and what are their probabilities. The acoustic models and language models are used in decoding for searching the recognition hypothesis that fits best to the models. Recognition output can then be used in various applications [7].

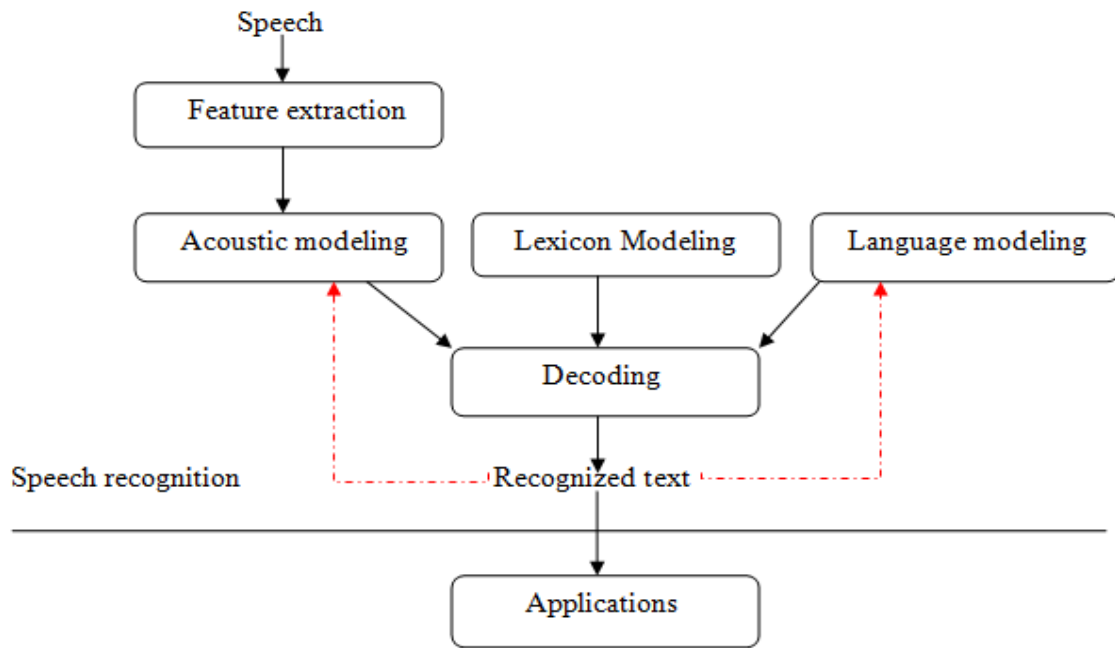


Figure 1. 1 The Main Components of an Automatic Speech Recognition System [8]

1.2 Classification of Automatic speech recognition

Speech recognition systems can be categorized based on different parameters important of which are based on speaking Mode (isolated or discrete and continuous speech), based on enrollment (speaker-dependent and speaker-independent),

Based on Vocabulary size (small, medium and large), based on Language Model (context-sensitive and context in-sensitive) and based on speaking style (Dictation and Spontaneous speech).Based on speaking mode (utterance) speech is either discrete or continuous.

Discrete speech consists of isolated words that are separated by silences [9]. The advantage of discrete speech is that word boundaries can be set exactly while with continuous speech; words will be spoken without silence [9].

The other category is based on the speaker enrollment it is either speaker dependent or independent. Speaker dependent systems are designed for a specific speaker [10]. They are generally more accurate for the particular speaker, but much less accurate for others speakers.

These systems are usually easier to develop, cheaper and more accurate than speaker dependent, but not as flexible as speaker adaptive or speaker independent systems. Speaker independent models are designed for variety of speakers.

It recognizes the speech patterns of a large group of people. This system is most difficult to develop, most expensive and has less accuracy than speaker dependent Systems. However, they are more flexible. Based on the language model we want to apply speech is either context-sensitive or context insensitive. More general language models approximating natural language are specified in what is known as a context sensitive grammar.

Speech recognition systems that increase their accuracy by anticipating or limiting what can be said at any given time are referred to as context-sensitive recognition systems whereas systems that allow users to say anything, anytime without any constraint are called context insensitive recognition systems. Context-sensitive recognition systems are more difficult to implement on computers than context-insensitive ones.

Based on speaking style automatic speech recognition is classified into either Read speech or Spontaneous speech. Continuous speech recognition systems allows the users to utter the message in a relatively or completely unconstrained manner. Such recognizers must be capable of performing well in the presence of all the co-articulatory effects. Developing continuous speech recognition systems is, therefore, the most difficult task. This is due to the following properties of continuous speech:

1. Word boundaries are unclear in continuous speech; and
2. co-articulatory effects are much stronger in continuous speech

1.3 Approaches in Automatic Speech Recognitions

The following are the modeling (approaches) which can be used in speech recognition process:

- **The Acoustic-phonetic approach:-**This approach is one of the earliest approaches to speech recognition and it is based on finding speech sounds and providing appropriate labels to these sounds [3]. It relies on explicit characteristics of speech sounds. In this approach the acoustic properties of phonetic unit is governed by straight forward rule to be understood by machine, but the acoustic properties of phonetic unit is not consistent since it is affected by different real world factors like speakers and neighboring phonetic unit [11]. In acoustic phonetic approach initially we have to do segmentation and labeling before proceeding to the next steps. Segmenting the speech signal into discrete (in time) region where the acoustic properties of the signal are representative of one (or possibly several phonetic unit or classes, and then attaching one or more phonetic labels to each segmented region according to the acoustic properties.
- **Pattern Recognition approach:-**also known as pattern-matching approach involves two essential steps namely, pattern training and pattern comparison [11]. The essential feature of this approach is that it uses a well formulated mathematical framework, robustness and invariance to different speech vocabularies in users, feature sets, tern comparison algorithms and decision rules. This property make it suitable for a wide range of speech unit and establishes consistent speech pattern representations, for reliable pattern comparison, from a set to labeled training samples via a formal training algorithm. The key idea of pattern recognition is to optimally extract patterns based on certain conditions and thereby separate the data into different classes. Moreover, pattern recognition approach performs classification without having any idea about the distribution of the measurements in different groups. There are two sub approaches under pattern recognition approach the template based approach and stochastic approach.
- i. **Template based approaches:-**Categorized under the pattern recognition approach “Template based approaches matching Unknown speech is compared against a set of per-recorded words (templates) in order to find the best match” [12]. As a reference a set of template is stored and read and the comparison for the unknown utterance is made against

the stored template. Recognition is carried out by selecting the category of the best matching pattern.

- ii. **Stochastic approach:**-Categorized under the pattern recognition approach. “Stochastic modeling entails the use of probabilistic models to deal with uncertain or incomplete information.” [12]. In speech recognition, uncertainty and incompleteness arise from many sources; for example, confusable sounds, speaker variability’s, contextual effects, and homophones words. Thus, stochastic models are particularly suitable approach to speech recognition.

The most popular stochastic approach today is hidden Markov modeling. A Hidden Markov Model is characterized by a finite state Markov Model and a set of output distributions. “It is a doubly stochastic process which has an underlying stochastic process that is not observable (hence the term hidden), but can be observed through another stochastic process that produces a sequence of observations.” [3]. The Hidden Markov model (HMM) is a very powerful mathematical tool for modeling time series data. It provides efficient algorithms for state and parameter estimation, and it automatically performs dynamic time warping for signals that are locally squashed and stretched. It can be used for many purposes other than acoustic modeling [13]. This specific study also use the stochastic approach and the well know speech recognition model Hidden Markov Model (HMM) is selected based on the above mentioned reasons.

- **The artificial intelligence approach:**-The artificial intelligence approach is “a hybrid of the acoustic phonetic approach and pattern recognition approach.” [11]. It took the advantage of the two approaches; the basic idea in this approach is to compile and integrate knowledge from a variety of knowledge source and to convey it on the problem at hand. The recognition procedure held in artificial intelligence approach is by mimic human’s capability of visualizing, analyzing, reasoning and making decision to applying it to computer on the measured acoustic features.

1.4 Statement of the Problem and Its Justification

The growth in Automatic speech recognition technology inspired different researchers to explore and investigate applications of automatic speech recognitions for different languages. [14] [15] Stated speech recognition is rich field of research for both practical and intellectual reasons. With this regard, the use of speech to communicate between human beings is inevitable of the primary choice and with no substitutes. A speech recognition system constructed for one language works only for that specific language. Most of commercial ASR systems work only for the most widely spoken languages in the world like English, French, Spanish, German, or Chinese. A person should speak one of these languages to be able to use speech recognition for any purpose. This makes speech recognition difficult to use for people that can't speak one of the above languages - which is quite a large number of population.

As a result of the tools for automatic speech recognition that are available freely for research purpose there are lots of researches have been done globally and locally on Automatic Speech Recognition including Afaan Oromo Automatic speech recognition which are developed in different times. Local researchers tried to alleviate the problem of speech in general and speech recognition specifically that the physical disabled person faces. To mention some Solomon [16] Kinfe [17], Zegaye [18], Martha [19], Hussien [20] have undertaken their studies on isolated word and on continuous speech recognition and related issues for specifically Amharic language using Hidden Markov Model (HMM), Artificial Neural Network (ANN) and Hybrid model.

But apart from the Amharic that is the working language of the Federal Democratic Republic of Ethiopia; Afaan Oromo is one of the languages predominantly spoken in Ethiopia and some neighboring countries like Kenya and some part of Somalia constituting more than 40 million people [21]. In addition to these Afaan Oromo is one of the languages used for instructional purpose from elementary to university programs and working language of the Regional State of Oromia, and used by many people in the country due to the large geographic coverage and number of speakers.

To the best of our knowledge there is only two research undertaken by [22] [23] towards designing isolated speech recognition for Afaan Oromo and one research by [15] A continuous, speaker independent Afaan Oromo speech recognizer.

The experimental results and technical detail explanations in [15] motivate us to conduct further research on the area of automatic speech recognition system for Afaan Oromo by further exploring the possible as a motivation of those previously done research works that are done on this research area and we are inspired to delve and work on the large vocabulary, continuous speech recognition for Afaan Oromo, using HMM.

The previous attempts on Afaan Oromo continuous read speech recognizer short words speech data, medium vocabulary size and speaker dependent test data set were used.

Therefore the aim of this study is to develop a recognizer which is speaker independent and speaker dependent for large vocabulary continuous read speech data and all Afaan Oromo phonemes were considered.

1.5 Research Questions

At the end of this study this questions are answered with the result obtained from the experiment:

- What are the challenges of developing continuous read speech for Afaan Oromo?
- What are the Acoustic model parameters that may improve recognition performance?
- Which language model performs better for Afaan Oromo speech recognition system?

1.6 Objective of the Study

1.6.1 General Objective

The general objective of this study is to explore the possibility of developing Large Vocabulary Continuous Speech Recognition System prototype for Afaan Oromo using HMM.

1.6.2 Specific Objectives

The specific objectives of the study are:

- I. Perform a comprehensive literature review of both related to automatic speech recognizer and the language under study.
- II. Identifying the best way of developing acoustic model, language model, and pronunciation dictionary for the recognizer.
- III. Collecting and preparing data sets for the system to train and test the models that will be built.
- IV. Develop a prototype Afaan Oromo automatic speech recognition system
- V. Evaluation of the system developed for experimentation
- VI. Draw conclusions based on the results obtained and forward recommendations directions for future studies

1.7 Scope and Limitations of the Study

We focused on checking the applicability and possibility of developing speaker independent and speaker dependent prototype for Afaan Oromo continuous speech recognizer. The system also integrated natural language processing and signal processing the speech signal in to computer understandable text. Language model, acoustic model and pronunciation dictionary are the main part of the recognizer.

Due to limitation of time and unavailability of Afaan Oromo corpus for the research, we are limited to work with 800 selected Afaan Oromo sentences uttered by 16 speakers. In which each speaker read 50 sentences.

In this study the dialect differences were not considered. This is mainly because the variation in dialect causes the differences in the performance of the recognizer and need to perform a lot of acoustic tasks.

Furthermore, we also encounter some difficulties during recording the speech data from selected individual due to lack of uttering correctly the given phrases and sentences. In addition to this the recording environment needs quit environment and in most cases it is difficult to get respondents in the specified areas.

1.8 Methodology of the Study

The study used different methodologies to realize the research work. These are the methods that we used in the study:-

Literature review

Comprehensive literature review were performed form books, journals, and the internet in order to investigate the underlying principles/ theories of the various approaches, techniques and tools that were employed in the research. Further literature review is also performed on the language Afaan Oromo and the phonetic characteristics of the language as well. Besides, literatures referring to similar research in the area of speech recognition were reviewed. This helped the researcher to understand the problem and to support the result in a scientific manner to be empirically evaluated.

Data collection methods

❖ Data preparation

Data preparation is the first task that should be addressed in speech recognition development process. The corpus size which consists of 800 sentences utterance of 16 speakers are prepared for this specific study by consulting linguistic domain expert. The corpus prepared has contain all phones of language under study. In this corpus necessary adjustment is made like removing symbols (@ # % '), converting numbers into equivalent text, removing punctuation marks, etc.

In this corpus one over ten of data is used for testing, the rest used for training.

One hour read speech data was collected, it was segmented and labeled into sentences by Praat freely available toolkits. We also prepare text corpus that is used for the language modeling and lexicon model from scratch.

❖ Development Tools and Technique

For the development of speech recognizer the selection of modeling tools is the most important step of the process. We have used Hidden Markov Model modeling technique that became the predominant technique for speech recognition. HMMs are at the heart of almost all modern speech

recognition systems especially the system which is used statistical method, although the basic framework has not been changed significantly in the last decade or more.

Prototype of large vocabulary continuous speech recognition system for Afaan Oromo were modeled using CMU sphinx open source speech recognition toolkit. This toolkit were preferred since different studies in this area had used the toolkit and achieved considerable results.

In addition to this, this toolkit is freely available for academic and research use. For language modeling we have used SRILM language modeling toolkit and for text normalization and preparation we have also used Python and Perl codes. The audio file is segmented into sentences using PRAAT tool. Notepad++, Net Beans IDE 8.1 and other software are used for text editing and for purposes where they needed.

❖ **Testing and Evaluation**

As clearly indicated in different literatures, the testing activities are one of the important tasks to be performed while doing an experiment. The recognition system is evaluated or tested for its accuracy and also we have tested the accuracy of recognizer. To test the recognition system, the CMU sphinx decoders are used and then the accuracy is reported on the results section with different reporting mechanisms.

Therefore, in this research the performance is evaluated against the test speech data set prepared using CMU sphinx decoding tool. Here the classification and other preparation of the training and test data sets were performed manually. Whereas the training received from the sphinx train was then taken to the decoder and the task of measuring the performance is done automatically using the appropriate decoding scripts.

1.9 Significance of the study

In a day to day activity peoples communicate through speech. It is a focus area now a day to make the communication between people and machine through speech. The communication between people is using continuous (spontaneous) speech therefore peoples need to communicate with machine by speech like they do with people; this study as a contribution towards developing a system that enable human being to communicate with a machine using speech.

Like other languages speech recognition, Afaan Oromo speech recognition is also very helpful for handicapped Afaan Oromo speakers that means for users who have difficulty in using their hands to type, but are able to speak clearly. In addition, blind users can use this kind of system since they have difficulty in using keyboard and mouse to command and control computers. Other group of users that can get benefit from this kind of system is people whose eyes and hands are busy in performing other task. In general, it can be said that such system is helpful for any people who can speak Afaan Oromo.

There were some attempts of studying ASR using small vocabulary, but this research is done using large vocabulary. Therefore, this study has its own contribution on the applicability of Afaan Oromo speech recognition, since effectively broadening the application of speech recognition depends crucially on raising recognition performance for continuous and spontaneous speech. This study can be used as an input for future researches on Afaan Oromo Automatic Speech Recognition System.

1.10 Organization of the Thesis

The thesis is organized in six chapters of which are briefed in the following ways:

Chapter One introduces all of the contents of the rest of the chapters and it also describes the statement of the problem and its justification, objectives of the research General objective and specific objectives, methodologies that are used for completing the entire study, scope and limitation of the study.

Chapter Two is literature reviews on the ASR systems, types of ASR, speech recognition tasks, stochastic HMM model approaches, speech units at ASR, the components of ASR, related works that are done on Amharic language and Afaan Oromo Language.

Chapter Three mainly discuss about the general description of Afaan Oromo language in related to Automatic Speech recognition. Overview of the language, Afaan Oromo Alphabets and Orthography, Afaan Oromo phonemes and writing system of the language under study.

Chapter Four describes about CMU sphinx tool kit we used for the study and other tools used in this study for different purposes.

Chapter Five describes about ASR system architecture, experimentation prototype and discussion of recognition performance obtained after decoding.

Chapter Six includes conclusion and recommended tasks for the coming researchers in this area.

CHAPTER TWO

LITERATURE REVIEW

In this chapter we have been reviewed different literatures that are related to speech recognition system also the basics of speech recognition. It is classified in different sections that describe different concepts in speech recognition. In the first section we discovered basics of speech recognition and on the other sections we discussed about ASR types, tasks and etc.

2.1 Basics of Speech Recognition

The following terms are the basics for understanding speech recognition technology [24].

Utterance

An utterance is the vocalization (speaking) of a word or words that represent a single meaning to the computer. Utterances can be a single word, a few words, a sentence, or even multiple sentences.

Speaker

Speaker dependent systems are designed around a specific speaker. They generally are more accurate for the correct speaker, but much less accurate for other speakers. They assume the speaker will speak in a consistent voice and tempo. Speaker independent systems are designed for a variety of speakers. Adaptive systems usually start as speaker independent systems and utilize training techniques to adapt to speaker to increase their recognition accuracy.

Vocabularies

Vocabularies (or dictionaries) are lists of words or utterances that can be recognized by the SR system. Generally, smaller vocabularies are easier for a computer to recognize, while larger vocabularies are more difficult. Unlike normal dictionaries, each entry doesn't have to be a single word. They can be as long as a sentence or two.

Accuracy

The ability of a recognizer can be examined by measuring its accuracy – or how well it recognizes utterances. This includes not only correctly identifying an utterance but also identifying if the

spoken utterance is not in its vocabulary. Good ASR systems have an accuracy of 98% or more! The acceptable accuracy of a system really depends on the application.

Training

Some speech recognizers have the ability to adapt to a speaker. When the system has this ability, it may allow training to take place. An ASR system is trained by having the speaker repeat standard or common phrases and adjusting its comparison algorithms to match that particular speaker. Training a recognizer usually improves its accuracy. Training can also be used by speakers that have difficulty speaking, or pronouncing certain words. As long as the speaker can consistently repeat an utterance, ASR systems with training should be able to adapt.

2.2 Overview of Automatic Speech Recognition

Speech recognition is concerned with converting the speech waveform, an acoustic signal, into a sequence of words. Today's most practical approaches are based on a statistical modeling of the speech signal. This chapter focuses on the statistical methods used in state-of-the-art speaker-independent, continuous speech recognition. Some of the primary application areas of speech recognition technology are dictation, spoken language dialog and transcription systems for information retrieval from spoken documents [3].

The speech recognition problem we have to solve is, someone produces some speech and we have to have a system that automatically translates this speech into a written transcription. To solve this problem among different approaches we can use statistical approach. From a statistical point of view, speech is assumed to be generated by a language model which provides estimates of $P(W)$ for all possible word strings $W = (w_1, w_2, w_3 \dots w_i)$, and an acoustic model represented by a probability density function $p(O|W)$ encoding the message W in the signal O . The goal of speech recognition is generally defined as finding the most likely word sequence given the observed acoustic signal [3].

The main components of a generic statistical speech recognition system are show in Figure 2.1 along with the requisite knowledge sources (speech and textual training materials and the pronunciation lexicon) and the main training and decoding processes. The acoustic and language

models resulting from the training procedure are used as knowledge sources during decoding, after feature analysis has been carried out from speech data by feature extraction (preprocessing). The rest of this chapter is devoted to discussing these main constituents and knowledge sources.

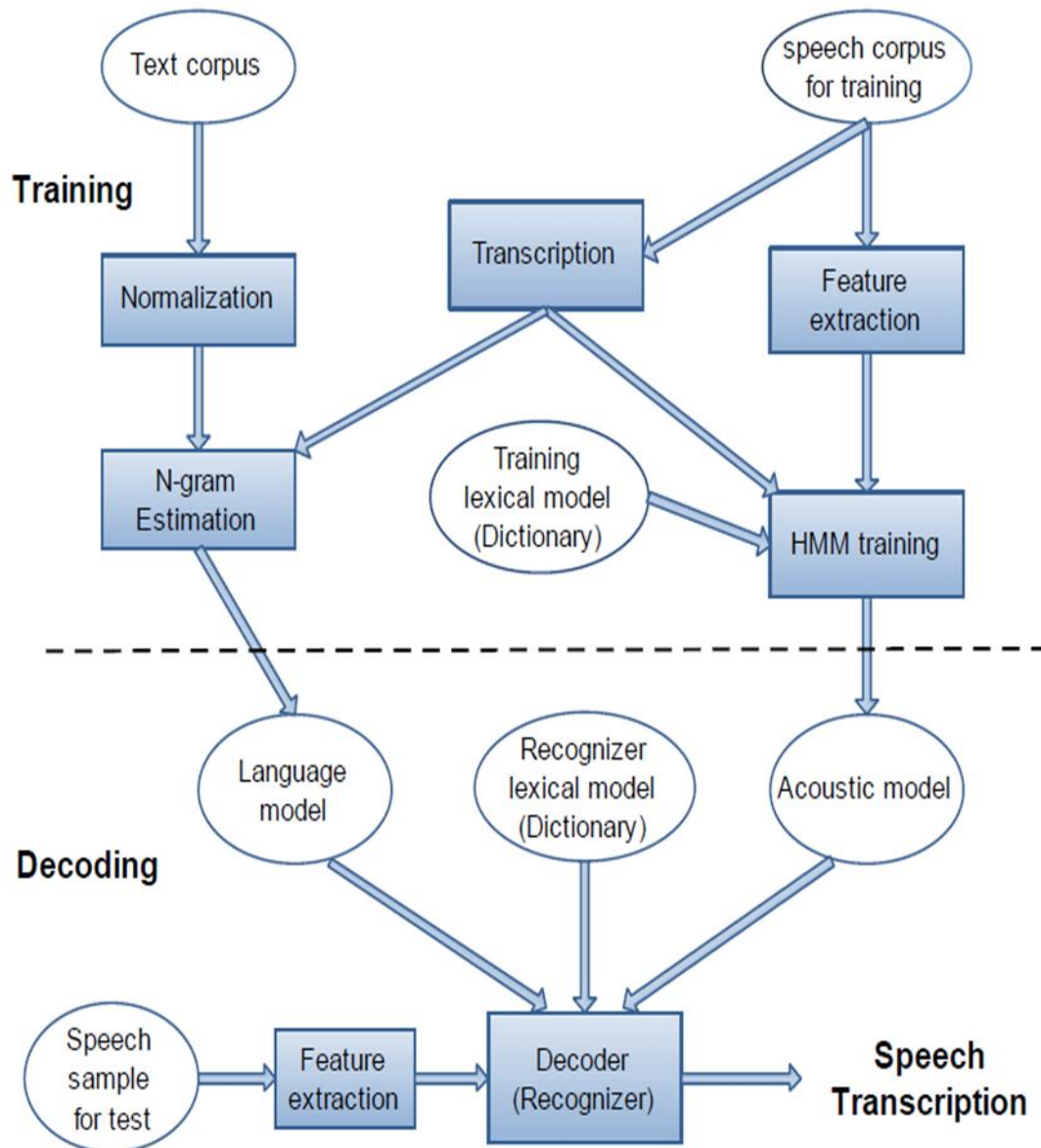


Figure 2. 1 Architecture of an ASR system based on statistical approach, adapted from [13]

The literatures reveal that the components of speech recognition are categorized into three. The acoustic model, lexical model and language models are the well-known components of automatic speech recognition. However, the continuous speech waveform is first converted to a sequence of

equally spaced discrete parameter vectors. The assumption is the speech vector is being stable, although this is not strictly true it is a reasonable approximation. Then by applying different parameters the extraction is done using feature extraction methods. Automatic Speech Recognition (ASR) systems have certain features which are composed of two parts: training and decoding. To train an ASR we need a set of training waveforms and their transcripts in order to build the acoustic, lexical, and language models.

The acoustic model captures statistics on phoneme realization, the lexicon contains mappings of words to their underlying acoustic units, and the language model defines the likelihood of one word following another. The other part in automatic speech recognition is decoding considered as the heart of automatic speech recognition which search for the most likely word sequence given the observed features extracted from the speech signal. This is commonly referred to as decoding or recognizing the speech signal.

2.3 Speech Recognition Tasks

According to Markowitz which is pointed on the work of [8] [19] emphasizing speech recognition system that plays the role of human-computer interface, indicated that such system performs three primary tasks: preprocessing, recognition and communication.

2.3.1 Preprocessing

Speech is an analog signal “a continuous flow of sound waves and silence”. This form cannot be processed directly by a speech recognizer. Thus, there is a need for preprocessing the speech signal. Preprocessing is the conversion of spoken input into the form that is acceptable by the recognizer. Specifically, it deals with digitizing analog speech and converting the speech wave form to some type of parametric representation.

Analog to digital conversion is done using analog to digital converters [19] whereas conversion to parametric representation is done by digital signal processing front ends. According to [25] signal processing is stated that in which features are extracted from speech waveform, and parameter estimation, in which model parameters are iteratively updated using the training corpus and the pronunciation dictionary.

2.3.2 Recognition

In this step which the system identifies what has been said after following preprocessing. There are three approaches to automatic speech recognition. These are: **acoustic-phonetic**, **pattern recognition** and **artificial intelligence** [19].

Acoustic Phonetic

Acoustic-phonetic approach is based on the theory of acoustic phonetics. This theory “postulates that there exist finite, distinctive phonetic units in spoken language and that the phonetic units are broadly characterized by a set of properties that are manifest in the speech signal, or its spectrum, over time.” It has two steps (segmentation and labeling, and word level recognition) besides the preprocessing step which is common to all other approaches.

Pattern Recognition

Pattern recognition approach to speech recognition, unlike acoustic-phonetic approach, is the one in which the speech patterns are used directly without explicit feature determination and segmentation. This method involves two steps: training of speech patterns and recognition of the pattern via pattern comparison.

During training phase speech knowledge is brought into the system. The idea behind the training phase is that if enough versions of a pattern to be recognized are included in the training set that is provided to the algorithm, the training algorithm should be able to adequately characterize the acoustic properties of the pattern. In the second step – pattern recognition – unknown speech or the speech to be recognized is compared with each possible pattern learned in the training phase and classified according to the goodness of match of the patterns.

Artificial Intelligence

AI approach is a hybrid of acoustic-phonetic and pattern recognition approaches since it exploits ideas and concepts of both methods. This approach attempts to simulate the way a person applies his intelligence in visualizing, analyzing and making a decision on a measured acoustic features. To do this, the approach requires different knowledge. In fact, the basic idea of AI approach is to “compile and incorporate knowledge from a variety of knowledge sources and

to bring it to bear on the problem at hand.” The knowledge required by AI approach is acoustic, lexical, syntactic, semantic and pragmatic. Some of the techniques used in this approach are the use of expert system for segmentation and labeling, learning and adapting over time, and the use of artificial neural network (ANN) for learning the relationship between phonetic events and all known inputs as well as for discrimination between similar sound classes.

2.4 Automatic Speech Recognition Types

Speech recognition systems can be categorized based on different parameters, some of the parameters and types of Automatic speech recognizers depending on these parameters are given below:

Based on Speaking Mode: Isolated (discrete) and continuous speech

Isolated (Discrete) speech recognition systems are systems that require the speaker to pause briefly between words. As it is explained by Markowitz [26], Speech is said to be continuous when it is uttered as a continuous flows of sounds with no inherent separations between them and speech recognition system developed using these type of speech is referred to as continuous speech recognition.

Based on Enrollment: Speaker-dependent and speaker-independent

Speaker-dependent system uses speech samples from the target speaker to learn the model parameters of the speaker’s voice. Speaker independent systems are designed to be used by any users who want to use them with no enrollment. This is also planned to be used for this study.

Based on Vocabulary size: small, medium and large

The number of words in the vocabulary is a constraint that makes a speech recognition system small, medium or large. As a rule of thumb, small vocabulary systems are those which have a vocabulary size in the range of 1-99 words; medium, 100-999 words; and large, 1000 words or more [27]. Large vocabulary speech recognition systems perform much worse compared to small vocabulary systems due to different factors such as word confusion that increases with the number of words in the vocabulary. For small vocabulary recognizers, each word can be modeled. However, it is not possible to train acoustic models for thousands of words separately because we

cannot have enough training speech and storage for parameters of the speech that is needed. The development of large vocabulary recognizers, therefore, requires the use of sub-word units.

Based on Speaking Style: Read speech and Spontaneous speech

Read speech is a speech which is made ready by the form of script and reader inserts false pauses between words while reading the text. If compared with spontaneous speech these speeches are more fluent and have less non speech events like filled pause, repetitions, hesitation and others. We can use these speech data for the development of speech recognizer; therefore we can say it is read speech recognizer [28].

Presently the art of state speech recognition is grown to the various domain of recognizing audio speeches, video speeches, telephone and the like.

2.5 Signal Processing and Feature Extraction

2.5.1 Signal Processing

The signal processing sub-system works on the speech signal to reduce the effects of the environment (e.g., clean vs. noisy speech), the effects of the channel (e.g., cellular/land-line phone versus microphone). The feature extraction sub-system parameterizes the speech waveform so that the relevant information (the information about the speech units) is enhanced and the non-relevant information (age-related effects, speaker information, and so on) is mitigated [29].

There are different methods that can be used to extract parameters of a speech, (1) **production-based** (2) **perception based**, by simulating the effect that the speech signal has on the speech perception system (3) **Signal based**, method to describe the signal in terms of its fundamental components.

Regardless of the method employed to extract features from the speech signal, the features are usually extracted from short segments of the speech signal. This approach comes from the fact that most signal processing techniques assumes the vocal tract as stationary, but speech is non-stationary due to constant movement of the articulators during speech production. However, due to the physical limitations on the movement rate, a segment of speech sufficiently short can

be considered equivalent to a stationary process. This approach is commonly known as short-time analysis **Signal based Analysis**.

2.5.2 Feature Extraction

Features extraction in ASR is the computation of a sequence of feature vectors which provides a compact representation of the given speech signal [29]. It is mainly performed in three main stages. The first stage is called the speech analysis or the acoustic front end, which performs spectra temporal analysis of the speech signal and generates raw features describing the envelope of the power spectrum of short speech intervals. The second stage compiles an extended feature vector composed of static and dynamic features. Finally, the last stage transforms these extended feature vectors into more compact and robust vectors that are then supplied to the recognizer.

Feature Extraction is the most important part of speech recognition since it plays an important role to separate one speech from other [10]. Because every speech has different individual characteristics embedded in utterances. These characteristics can be extracted from a wide range of feature extraction techniques proposed and successfully exploited for speech recognition task. Extracted feature should meet some criteria while dealing with the speech signal such as:

- It should be easy to measure extracted speech features.
- It should not be susceptible to imitation.
- It should show little fluctuation from one speaking environment to another.
- It should be stable over time.
- It should occur frequently and naturally in speech.

The purpose of feature extraction stage is to extract the speaker-specific information in the form of feature vectors. In addition to represent the speech wave forms in more compact manner. There are different techniques (methods) in order to extract feature from speech.

2.5.3 Production Based Analysis

The speech production process can be described by a combination of a source of sound energy modulated by a transfer (filter) function. This theory of the speech production process is usually referred to as the source-filter theory of speech production. The transfer function is

determined by the shape of the vocal tract, and it can be modeled as a linear filter. However, the transfer function is changing over time to produce different sounds. The source can be classified into two types. The first one is which is responsible for the production of voiced sounds (e.g., vowels, semivowels, and voiced consonants). This source can be modeled as a train of pulses. The second one is related to unvoiced excitation. In this type, this source can be modeled as a random signal.

2.5.4 Perception Based Analysis

Perception-based analysis uses some aspects and behavior of the human auditory system to represent the speech signal. Given the human capability of decoding speech, the processing performed by the auditory system can tell us the type of information and how it should be extracted to decode the message in the signal. Two methods that have been successfully used in speech recognition from this method of analysis are; Mel-Frequency Cepstrum Coefficients (MFCC) and Perceptual Linear Prediction (PLP).

Mel-Frequency Cepstrum Coefficients (MFCC)

Mel Frequency Cepstrum Coefficients (MFCC):-The most prevalent and dominant method used to extract spectral features is calculating Mel Frequency Cepstral Coefficients (MFCC). MFCCs are one of the most popular feature extraction techniques used in speech recognition based on frequency domain using the Mel scale which is based on the human ear scale. MFCCs being considered as frequency domain features are much more accurate than time domain features.

The Mel-Frequency Cepstrum Coefficients is a speech representation that exploits the nonlinear frequency scaling property of the auditory system. This method warps the linear spectrum into a nonlinear frequency scale, called Mel. The Mel-scale attempts to model the sensitivity of the human ear and it can be approximated by the following formula:

$$B(f) = 1125 \ln \left(1 + \frac{f}{700} \right), \dots\dots\dots 2.1$$

For frequency f , the scale is close to linear for frequencies below 1 kHz and is close to logarithmic for frequencies above 1 kHz. MFCCs which are implemented for this study are often used in many other speech recognition systems.

Body Linear Predictive Codes (LPC):- It is desirable to compress signal for efficient transmission and storage. Digital signal is compressed before transmission for efficient utilization of channels on wireless media. For medium or low bit rate coder, LPC is most widely used [19]. The LPC calculates a power spectrum of the signal. It is used for formant analysis [30]. LPC is one of the most powerful speech analysis techniques and it has gained popularity as a formant estimation technique.

Perceptual Linear prediction (PLP):-The Perceptual Linear Prediction PLP model developed by Hermansky. PLP models the human speech based on the concept of psychophysics of hearing [29]. PLP discards irrelevant information of the speech and thus improves speech recognition rate. PLP is identical to LPC except that its spectral characteristics have been transformed to match characteristics of human auditory system.

2.6 Acoustic Models

The acoustic model determines what sounds will be produced when a given string of words is Uttered. Thus for all possible combinations of word strings $\mathbf{W}$ and observation sequences $\mathbf{O}$ the Probability $P(\mathbf{O}|\mathbf{W})$ must be available.

$$P(W|o) \quad \text{-----} \quad 2.2$$

$$W = \operatorname{argmax}$$

This number of combinations is just too large to permit a lookup; in the case of continuous speech it's even infinite. The total process modeled involves the way the speaker pronounces the words of $\mathbf{W}$, the ambience (room) noise, the microphone placement and characteristics and the acoustic processing performed by the signal processing front end.

Unfortunately, unless there is some limit on the duration of the utterances and a limited number of observation symbols, this equation is not directly computable since the number of possible

observation sequences is totally infinite. But Bayes formula gives:

$$P(W | O) = (P(W)P(O | W)) / (P(O)) \dots\dots\dots 2.3$$

From the above formula, $P(\mathbf{W})$, is called the language model, which is the probability that the word string $\mathbf{W}$ will be uttered and $P(\mathbf{O}|\mathbf{W})$ is the probability that when word string $\mathbf{W}$ is uttered the acoustic evidence $\mathbf{O}$ will be observed, which is called the **acoustic model**. The probability $P(\mathbf{O})$ is usually not known but for a given utterance it is of course just a normalizing constant and can be ignored. Thus to find a solution to formula (2.2) we have to find a solution to:

$$W_{argmax} P(W)P(O | W) \dots\dots\dots 2.4$$

The most frequently used model these days is the Hidden Markov model but other models are possible, for instance artificial neural networks [13].

Acoustic model is the main component for an ASR and it accounts for most of the computational load and performance of the system. It is used to link the observed features of the speech signals with the expected phonetics of the hypothesis sentence and used for detecting the spoken phoneme. Its creation involves the use of audio recordings of speech and their text scripts and then compiling them into a statistical representation of sounds which make up words. Acoustic models represent sub-word units, such as phonemes, as a finite-state machine in which states model spectral structure and transitions model temporal structure.

2.6.1 Hidden Markov Model (HMM)

The core of pattern matching speech recognition approach is a set of statistical models representing the various sounds of the language to be recognized. Since speech has sequential structure and can be encoded as a sequence of spectral vectors, the hidden Markov model (HMM) provides a natural framework for constructing such models.

HMM is a Markov chain plus emission probability function for each state. In the Markov model each state corresponds to one observable event. But this model is too restrictive, for a large number

of observations the size of the model explodes, and the case where the range of observations is continuous is not covered at all.

As described by Jurafsky, and et.al [31] an HMM is specified by a set of states Q , a set of transition probabilities A , a HMM set of observation likelihoods B , a defined start state and end state(s), and a set of observation symbols O , which is not drawn from the same alphabet as the state set

Q :

A Hidden Markov model can be defined by the following parameters:

- $S = \{s_1, s_2, \dots, s_N\}$: A set of states (usually indicated by i, j) is a state that the model is in at a particular point in time t . it will be indicated by st , thus $st = i$ means that the model is in state i at time t .
- $A = a_{11} \ a_{12} \ \dots \ a_{ij}$: A transition probability A , each a_{ij} representing the probability of moving from state i to state j .
- $O = o_1 \ o_2 \ \dots \ o_N$: A set of observations, each one drawn from a vocabulary $V = v_1, v_2 \dots v_v$
- $B = b_i(o_t)$: A set of observation likelihoods: also called emission probabilities, each expressing the probability of an observation o_t being generated from a state i .
- $\pi = \pi_1, \pi_2, \dots, \pi_N$: An initial probability distribution over states : π_i is the probability that s_i is s starting state.
- $\lambda = (A, B, \pi)$: Full HMM

HMM Problems and Their Solution

HMM three basic problems are Evaluation, Decoding and Training. The next topics will discuss these three problems and their solution.

Problem1 (Computing likelihood): Given an HMM $\lambda = (A, B, \pi)$ and an observation sequence O , determine the likelihood $P(O|\lambda)$?

Problem2 (Decoding): Given an observation sequence O and an HMM $\lambda = (A, B, \pi)$, discover the best hidden state sequence Q ?

Problem3 (Learning): Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B .

Solution to Problem 1 (computing likelihood): The Forward Algorithm

The forward algorithm is a kind of dynamic programming algorithm, an algorithm that uses a table to store intermediate values as it builds up the probability of the observation sequence. The forward algorithm computes the observation probability by summing over the probabilities of all possible hidden state paths that could generate the observation sequence, but it does so efficiently by implicitly folding each of these paths into a single forward frame

Each cell of the forward algorithm frame $\alpha_t(j)$ represents the probability of being in state j after seeing the first t observations, given the model λ . The value of each cell $\alpha_t(j)$ is computed by summing over the probabilities of every path that could lead us to this cell. Formally, each cell expresses the following probability:

$$\alpha_t(j) = P(o_1, o_2, \dots, o_t, q_t = s_j | \lambda) \text{-----} 2.5$$

We compute this probability by summing over the extensions of all the paths that lead to the current cell. For a given state s_i at time t , the value $\alpha_t(j)$ is computed as:

$$\alpha_t(j) = \left[\sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(o_t) \text{.....} 2.6$$

The three factors that are multiplied equation 2.6 for extending the previous paths to compute the Viterbi probability at time t are: $\alpha_{t-1}(i)$ The **previous forward path probability** from the previous time step α_{ij} The **transition probability** from previous state q_i to current state q_j $b_j(o_t)$ The **state observation likelihood** of the observation symbol o_t given the current state j . We can define the forward algorithm using a statement of the definitional recursion:

✓ Initialize

$$\alpha_1(i) = \pi_i b_i(o_1) 1 \leq i \leq N \dots\dots\dots 2.7$$

✓ Recursion (since states 0 and N are non-emitting)

$$\alpha_t(j) = \left[\sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(o_t) \quad 2 \leq t \leq T, 1 \leq j \leq N \dots\dots\dots 2.8$$

✓ Termination

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i) \dots\dots\dots 2.9$$

Solution to HMM Problem2 (Decoding): Viterbi algorithm

Decoding problem deals with, given a model and an observation sequence, finding the most likely or optimal state sequence in the model that produced the observation sequence. Since the state sequence is hidden in an HMM. Thus, to solve the problem it is possible to produce the state sequence that has the highest probability of being taken while generating the observation sequence. To do this we can use viterbi algorithm, which is a modification of forward algorithm. Instead of summing probabilities that came together as in the forward algorithm, in viterbi we need to choose and remember the maximum probability.

The viterbi algorithm has one component that the forward algorithm does not have: **back pointers**. This is because while the forward algorithm needs to produce observation likelihood, the Viterbi algorithm produces a probability and also the most likely state sequence [1]. We compute this best state sequence by keeping track of the path of hidden states that led to each state.

We want to find the state sequence $Q=q_1 \dots q_T$, such that:

$$Q = \underset{Q'}{\operatorname{argmax}} P(Q' | O, \lambda) \dots\dots\dots 2.10$$

Similar to computing the forward probabilities, but instead of summing over transitions from incoming states, compute the maximum:

$$\delta_t(j) = (\max_{1 \leq i \leq N} \delta_{t-1}(i) a_{ij}) b_j(o_t) \dots\dots\dots 2.11$$

Solution to Problem3: The Forward – Backward Algorithm (Baum-Welch algorithm)

The third problem of HMM is the learning (training) problem in which, given the model and an observation sequence, we attempt to adjust the model parameters to maximize the probability of generating the observation sequence.

Rabiner and Juang [1] supposed this problem is the most difficult problem since there is no known analytical method to solve for the model parameters that maximizes the probability of the observation sequence. An iterative procedure is used to solve this problem. One iterative procedure that is used to solve this problem is the forward – backward algorithm, which is also called Baum Welch algorithm. Using an initial parameter instantiation, the forward-backward algorithm iteratively re-estimates the parameters and improves the probability that given observations is generated by the new parameters.

Here there are three parameters need to be re-estimated:

- Initial state distribution: π_i .
- Transition probabilities: $a_{i,j}$.
- Emission probabilities: $b_i(o_t)$

Re-estimating the transition probabilities

Here we have to solve, what is the probability of being in state s_i at time t and going to state s_j , given the current model and parameters?

$$\xi_t(i, j) = P(q_t = s_i, q_{t+1} = s_j \mid O, \lambda) \dots\dots\dots 2.18$$

Let $\xi(i, j)$ be a probability of being in state i at time t and at state j at time $t+1$, given λ and O ;

$$\begin{aligned} \xi(i, j) &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O \mid \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \dots\dots\dots 2.19 \end{aligned}$$

The perception behind the re-estimation equation for transition probabilities is:

$$\hat{a}_{i,j} = \frac{\text{expected number of transitions from state } s_i \text{ to state } s_j}{\text{expected number of transitions from state } s_i} ;$$

$$\hat{a}_{i,j} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \sum_{j'=1}^N \xi_t(i, j')} \dots\dots\dots 2.20$$

The above equation can be modified as:

$$\hat{a}_{i,j} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \dots\dots\dots 2.21$$

Re-estimating Initial state probability

Initial state distribution is the probability that s_i is a starting state.

Re-estimation is π : 1 at time s state in times of number expected.

$$\hat{\pi}_i = \gamma_1(i) \dots\dots\dots 2.22$$

Re-estimation of Emission probabilities

$$\hat{b}_i(k) = \frac{\text{expected number of times in state } s_i \text{ and observe symbol } v_k}{\text{expected number of times in state } s_i}$$

$$\hat{b}_i(k) = \frac{\sum_{t=1}^T \delta(o_t, v_k) \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)} \dots\dots\dots 2.23$$

2.7 Lexical Models

Lexicon is developed to provide the pronunciation of each word in a given language [32]. Through lexical model, various combinations of phones are defined to give valid words for the recognition. The type of speech unit determines the pronunciation dictionary to be prepared .i.e the lexicon prepared for phones are not the same as for the speech unit syllable.

A pronunciation dictionary can be classified as a canonical or alternative on the basis of the pronunciations it includes. A canonical pronunciation dictionary includes only the standard phone or other sub-word sequence assumed to be pronounced in read speech. It does not consider pronunciation variations such as speaker variability, dialect, or co-articulation in conversational speech. On the other hand, an alternative pronunciation dictionary uses the actual phone or other

sub-word sequences pronounced in speech. In an alternative pronunciation dictionary, various pronunciation variations can be included.

2.8 Language Models

Language modeling is a task assigning a probability to a given sequence of words. It provides word-level structure for a language and applicable for many speech and natural language application such as automatic speech recognition (ASR), statistical machine translation (SMT) and optical character recognition (OCR). By using Language modeling automatic speech recognition (ASR) benefited in two ways. It reduces the search space for searching and providing contextual information [13]. Since language model ignore unlikely candidate, and thus the search problem become feasible.

With regarding to the second concept in speech there is a concept homonym means their pronunciation is exactly the same. Language model provide us contextual information to distinguish homonyms most of the time we can come up with a clear distinction between homonyms by applying language modeling [13]. To assign probability to given input word string the most popular language model is n-gram model.

Based on the number of word used for conditioning the n-gram language model is classified into three namely uni- gram, bi-gram and tri-gram. If one word is used for conditioning to determine the probability of word it is called uni-gram, in other case if we are considering two words for conditioning to determine the probability of word it is bi-gram and if we are considering three words for conditioning to determine the probability it is tri-gram.

Language model predicts the next set of words, and controls which models are hypothesized. It is the single largest component consists of parameters and developed for detecting the connections between the words in a sentence with the help of pronunciation dictionary [12]. ASR systems utilize n -gram language models to guide the search for correct word sequence by predicting the likelihood of the n th word on the basis of the $n-1$ preceding words. N -gram language model are easy to build as well as we need simple count of the word n -gram event from the training set.

But there is a problem in n -gram model for the unseen word in the training set the probability is zero this affects the other word which appear in the training set is also zero, since the product of

the two is zero. In order to solve the problem of n-gram model there is an introduction of several smoothing techniques among them several of them work well in ASR applications. “The fundamental idea of smoothing technique is to reserve (subtract) some small probability mass from the relative frequency estimate of the probabilities of seen event, and to redistribute this probability to unseen events” [13]. Smoothing methods differs according to how much is subtracted out (discounting) and how it’s redistributed (back-off).

2.9 Speech Units in Automatic Speech Recognition

A crucial issue for acoustic modeling is the selection of the speech units that represent the acoustic and linguistic information for the language. Speech is represented by using the various form of speech unit in automatic speech recognition. Ranging from smallest unit phones to the largest which is the word.

Word is the natural unit of speech because ultimately it is word that one is trying to recognize. Moreover, word units have their acoustic representation well defined. Acoustic variability occurs mainly in the beginning and end of a word i.e. at word boundaries. However, using word as a speech unit in large vocabulary continuous speech recognition (LVCSR) introduces several serious problems. Since each word has to be trained individually and there cannot be any sharing of parameters among words, one has to have a very large training set so that all words in the vocabulary are adequately trained.

The second problem lies with memory requirement which grows linearly with the number of words. Hence word models are not practical for large vocabulary continuous speech recognition (LVCSR) systems. But word models have been successfully used for limited vocabulary ASR [33]. A smaller or sub-word unit is the next choice because one has to have sharing of parameters across models in order to save computing resources.

The most popular sub-word units are the phones, as it is a well-known fact that the same phone in different words has different realizations. This is because the articulators cannot move from one position to another instantaneously. Hence the realization of a phone is strongly affected by its adjacent phones or in other words, phones are highly context dependent [33]. Some phones are

aspirated when they occur in the beginning of a word and the same phones are not aspirated when they occur at the end of a word.

Therefore the acoustic variability of basic phonetic units due to context is sufficiently large and not well understood in many languages. This problem of over-generalization can be overcome by employing phone-in-context. A context refers to immediate left and/or right neighboring phones. Phone-in-context can be either a left-context phone or a right-context phone or both. The third category is also known as tri-phone which includes both the left and right contexts. If two phones have the same identity but different left or right contexts, then they are considered as different tri-phones.

Tri-phone models are powerful sub-word models because they account for the left and right phonetic contexts. Tri-phones have been enormously successful in acoustic modeling of LVCSR systems [33]. Compared to word-model and phone-model, tri-phone model reduced word error rate (WER) by more than 50% [33].

However there are problems with tri-phones also. For any language, there are large numbers of tri-phones in the training set. This leads to memory wastage since a model is created for a tri-phone even if, that tri-phone is observed only once in the training set. A syllable is a larger unit than a phone since it encompasses two or more phone clusters. These phone clusters account for the severe contextual effects.

A syllable is composed of three parts: the onset, the nucleus and the coda or rhyme. The nucleus or central part does not have any contextual dependencies while the onset and the coda are still susceptible to some contextual effects. Syllables have long unit and they have the least context sensitive. The advantage of using syllables as a unit of training is that pronunciation variation is trained right into the acoustic model and does not need to be modeled separately in the dictionary. Syllable models also automatically capture co-articulation effects.

2.10 Related works

2.10.1 Review of Works done so far on Amharic ASR

Kinfe [17] developed sub-word based isolated word recognition systems for Amharic using HTK. The sub-word units used in the experiment are phones, triphones and CV-syllables. It considered 20 phones (out of 39) and 104 CV syllables, which are formed using the selected phones. Speech data of 170 words, which are composed of the selected sub-word units, have been recorded by 20 speakers (speech of 15 speakers for training and the remaining for testing).

Speaker dependent phone-based and triphone-based systems have an average recognition accuracy of 83.07% and 78% respectively. Phone-based and triphone-based speaker independent systems have an average recognition accuracy of 72% and 68.4% respectively. In addition, a comparison of the different sub-word units revealed that the use of CV syllables has led to relatively poor performance.

Zegaye [18] investigated the possibility of developing large vocabulary, speaker independent and continuous speech recognizer for Amharic language. The recognizer was developed using both phone-based and tri-phone based recognizers using HTK. For the training and testing of its recognizers, it used the speech data of 8000 sentences read by 80 people. In order to support the acoustic models, a bigram language model was constructed.

In addition, pronunciation dictionary was prepared and used as an input. Since the recognizer was meant to recognize large vocabulary and continuous speech, phonemes were opted as the basic unit of recognition. The best recognizer was a tri-phone based recognizer which has 76.2% word recognition accuracy.

Martha [19] explored the possibility of developing an Amharic speech input interface to command and control Microsoft Word. It required a speech recognizer and a communication interface between the recognizer and the application. 50 command words were selected from different menus (File, View, Insert, Tools, Table, Window, and Help), translated to Amharic and used to develop the prototype system.

To develop and test the required Amharic speech recognition system, speech data were recorded from 26 people (10 female and 16 male) in the age range of 20 to 35. 76.9% of the recorded data were used to train the recognizers and the remaining data were used for testing the performance of recognizers. To test the performance of the system, 18 randomly selected command words were given to 6 people (3 command words for each) and these people were asked to command Microsoft Word orally. The system performed 16 commands accurately and only two command words were wrongly recognized and thus Microsoft Word performed wrong actions.

Hussein and Gamback [20] developed an Amharic speaker independent continuous speech recognizers based on an HMM/ANN hybrid approach. The model was constructed at a context dependent phone level with the help of the CSLU Toolkit. It used part of the data (5000 sentences) recorded by Solomon *et al.* (2005). The recognizers were tested with a total of 20 sentences read by 10 speakers (2 sentences each) who also read the training data. When the same recognizer was tested for another ten speakers who were not involved in the training, the recognition rate degraded. The best result obtained with this test data was 74.28% word recognition accuracy.

Solomon [11] developed an Amharic speech corpus that can be used for various kinds of investigations in the development of ASR for Amharic. It explored various possibilities for developing a Large Vocabulary Speaker Independent Continuous Speech Recognition System for Amharic. The work assumed that, due to their highly regular Consonant Vowel (CV) structure, Amharic syllables lend themselves to be used as a basic recognition unit. Indeed, it has been able to show that syllable models can be used as a competitive alternative to the standard architecture that is based on triphone models.

The acoustic model of the syllable-based recognizer requires 15MB memory. Together with the language model and use of speaker adaptation, it has a word recognition accuracy of 90.43% on the 5,000 words evaluation test set at a speed of 2.4 minutes per sentence. While the acoustic model of the triphone-based recognizer requires 38MB memory and has a word recognition accuracy of 91.31% on the same test set at a speed of 3.8 minutes per sentence. Although this is the state-of-the-art recognition performance in Amharic, it still sees the room for improvement because the word recognition accuracy of ASR in the technologically favored languages is approaching to 100%.

Solomon [11] developed a multiple pronunciation model for Amharic speech recognition system based on the five dialects of Amharic language (Gonder, Gojam, Shewa, Menz and Wello). The research was based on Hidden Markov Model (HMM) and was guided by the assumption that incorporating variations of pronunciation from the Amharic dialects in ASRS can improve performance of Amharic Speech Recognition Systems.

The study used 731 and 129 utterances for training and testing respectively. The research found that using Amharic phone-based system with five emitting states multiple pronunciation dictionary together with set of acoustic model and language model obtained a word and sentence recognition accuracy of 85.54 %.

2.10.2 Review of Works done so far on Afaan Oromo ASR

Ashenafi [22] in his thesis entitled as A Speech Recognition System for Afaan Oromo explored the possibility of developing an automatic speech recognizer for Afaan Oromo. In his thesis the researcher considered isolated word speaker independent speech recognizer using Hidden Markov Model and an open source speech recognition toolkit called Sphinx4. For the purpose of achieving his objective the researcher prepared a speech corpus constructed using 50 Afaan Oromo words chosen by consulting the domain expert and uttered by 20 different people providing 1000 utterances making the entire dataset.

Accordingly the data set is classified in such a way that 2/3 is used for training and the remaining 1/3 for evaluation. Preprocessing and constructing of the acoustic model, the language model and the pronunciation dictionary with the help of sphinx4 and HMM was done. The performance of the recognizer is evaluated to test the recognition accuracy, transcription speed and type of errors occurred in decoding the test set. Accuracy tracker tool is used to test the performance of the system which is an integral part of the Sphinx system. Accordingly, 82.83 % and 81.081% word level accuracy is obtained for context dependent phoneme based models and context independent word based models, respectively.

Teferi [23] in his thesis entitled Speech Recognition for Afaan Oromo Using Hybrid Hidden Markov Models and Artificial Neural Networks, a hybrid HMM/ANN model is employed to see the effectiveness compared to the more common HMM. The researcher used CSLU hybrid

toolkit for developing and testing the hybrid ASR system 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 Afaan Oromo word are selected and 29 phonemes of the language are considered during the selection. To make the recording easy, the words are organized to form sentences. For this 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 98.11% was achieved.

Kassahun [15] in his thesis entitled as A continuous, speaker independent Afaan Oromo speech recognizer the researcher used 70 selected Afaan Oromo long words, phrase and simple sentence uttered by 30 selected peoples from different age group and sex constituting of 2100 utterances. These Afaan Oromo selected words, phrases and simple sentences are selected in consultation of the domain experts. The researcher used 2/3 of the prepared data for training and 1/3 for testing. In his thesis the researcher considered continuous speaker independent speech recognizer using Hidden Markov Model and an open source speech recognition toolkit called Sphinx4.

Performance evaluation is performed using test data sets and the recognizer performance is found to be 68.514% with sentence accuracy of 28% for continuous Afaan Oromo speech and a phoneme based trigram performance of 89.459% with sentence accuracy of 42% achieved.

CHAPTER 3

THE AFAAN OROMO LANGUAGE

In this chapter we will discuss about the general description of Afaan Oromo language in related to Speech recognition application by dividing into different section. The first section 3.1 will discuss about the overview of the language. Section 3.2 discuss about Afaan Oromo Alphabets and Orthography. Section 3.3 discuss about the basics of Afaan Oromo phonetics. Section 3.4 discuss about the writing system of the language under study.

3.1 The Overview of Afaan Oromo Language

The great Oromo people has their own culture, tradition, costumes and language. The Oromo's live in East Africa, particularly in Ethiopia, and parts of Kenya and Somalia. In Ethiopia, Geographically the largest federal states were Oromia. The Oromo people are the largest ethnic group in Ethiopia and account for more than 40% of the population. They can be found all over Ethiopia and particularly in Wollega, Shoa, Illubabour, Jimma, Arsi, Bale, Hararghe, Wollo, Borana, and the southwestern part of Gojjam. Afaan Oromo is an Afro-Asiatic language, and the most widely spoken of the Cushitic family. It is spoken as a first language by more than 40 million Oromo and neighboring peoples in Ethiopia and Kenya [21].

According to Tilahun [34] about 95 percent of Oromo speakers live in Ethiopia, mainly in Oromia region. Besides native speakers, a number of members of other ethnicities who are in contact with the Oromo's speak Afaan Oromo as a second language. For instance, the Omotic-speaking Bambassi and the Nilo-Saharan-speaking Kwama in north-western Oromia are the notables [21]. In Somalia there are also about some speakers of the language. According to Tilahun, in Kenya, the Ethnologue also lists 322,000 speakers of Borana and Orma, two languages closely related to Ethiopian Oromo. Currently, the above figure increased more than the above figures indicated. In addition to this, in Africa, it is the language with the third (3rd) most speakers, after Arabic and, Swahili.

Afaan Oromo had remained essentially a well-developed oral tradition until the early 1970's when the Oromo Liberation Front (OLF) began to use it as an official language in the liberated areas. The Front adopted the Latin script as its official alphabet, too [34].

The adoption of a script for Afaan Oromo had been a burning issue. In the 1970's both Sabeen and Latin scripts were suggested. Until 1974 when Mengistu's ruthless regime came to power, writing Afaan Oromo in any script had been banned officially. Although Mengistu's regime lifted the ban and reluctantly allowed the use of the Sabeen script, it continued to pay only lip service to the development of Afaan Oromo. For instance, the regime made the teaching of Afaan Oromo illegal at any level in its school system.

About five months after the collapse of Mengistu's regime in May 1991, the OLF convened a meeting of Oromo intellectuals on November 3, 1991. The purpose of the meeting was to adopt the Latin script the OLF had been using or suggest an alternative. Over 1000 Oromo intellectuals met in the Parliament Building at Arat Kilo, Finfinne (Addis Ababa).

The exact time when the Latin alphabet started being used for Afaan Oromo writing was not well known, but on November 3, 1991 it adopted as official alphabet of Afaan Oromo on. Now it is language of public media, education, social issues, religion, political affairs, and technology.

3.2 Afaan Oromo Alphabets and Orthography

Afaan Oromo uses Latin script or characters (Roman orthography). The script of Afaan Oromo is called ‘QUBEE’. Like in English and other languages, Afaan Oromo has vowels and consonants alphabets. Afaan Oromo vowels are represented by the five basic letters such as A, E, I, O, U. Afaan Oromo has 27 consonants including combination consonants. ‘Z’, ‘P’ and ‘V’ letters are not Afaan Oromo letters because there is no Oromo word written by use of either of them. But they are included by considering the fact of handling borrowed terms from other languages like English. For example: ‘Police’, ‘Piano’, ‘Television’, ‘video’, and etc. letter ‘Z’ was included in the alphabet as there are words which require the letter. For example: Ziqaya (gold), Zeeytuuna (guava), Azoole (river in Arsii), Zeekkara (Opera), Zalmaaxaya (mess), Waziiza (fire place or fire work) and Zawii (insanity) and etc.

Consonants, on the other hand, do not differ greatly from English, but there are few special combinations such as “**ch**” and “**sh**” (same sound as English), “**dh**” in Afaan Oromo is like an English “d” produced with the tongue curled back slightly and with the air drawn in so that a glottal stop is heard before the following vowel begins. Another Afaan Oromo consonant is “**ph**” made when with a smack of the lips toward the outside “**ny**” closely resembles the English sound of “gn”. We commonly use these few special combination letters to form words. For instance, **ch** used in **barbaachisaa** ‘important’, **sh** used in **shamarree** ‘girl’, **dh** use in **dhadhaa** ‘butter’, **ph** used in **buuphaa** ‘egg’, and **ny** used in **nyaata** ‘food’.

Qubee Afaan Oromoo (Afaan Oromo Alphabets)

Aa	Bb	Cc	CHch	Dd	DHdh	Ee	Ff	Gg	Hh	Ii
Jj	Kk	Ll	Mm	Nn	NYny	Oo	Pp	PHph	Qq	Rr
Ss	SHsh	Tt	TSts	Uu	Vv	Ww	Xx	Yy	Zz	

Table 3. 1 Afaan Oromo Alphabets

Afaan Oromo Vowels (Dubbachiftuu)

Afaan Oromo vowels are similar to that of English but sound differently. All vowels are pronounced basically the same way throughout Oromia.

All vowels pronounced in the similar way throughout every Afaan Oromo literature. These vowels pronounced in sharp and clear fashion which means, each and every word is pronounced strongly.

Afaan Oromo Consonants (Dubbifamtoota)

All Afaan Oromo consonants except the combination consonants or digraphs (“**Qubeewwaan Dachaa**”) like **CH**, **DH**, **NY**, **PH**, **TS** and **SH** have double consonant combinations if the syllable is stressed. Failure to make this distinction results in miscommunication. Examples: Bilisummaa, adda, malamaltummaa, damma, keessummaa, etc.

Some Afaan Oromo words are pronounced with the stress on the last syllable: for example: malamaltummaa, laggeen, gaarren. On the other hand, few words are stressed on the first syllable.

These words always have a combination consonant: e.g nyaadhu, dhayi, nyaara nyaapha (foreigner).

3.3 Basics of Afaan Oromo Phonetics

Phonetics is the study of sounds of a language that produce a word. These sounds are called phonemes. Languages vary in the phonemes they have. According to [35] [36] Afaan Oromo is a phonetic language, which means that it is spoken in the way it is written.

The Afaan Oromo phoneme inventory includes twenty-seven consonants and five vowels. Vowel length is contrastive, geminates (double consonants) are tolerated, diphthongs (double discrete vowels) do not occur, and consonant clusters are attested although highly restricted. The syllable structure of Oromo can be schematized as follows: CV (V) (C), where C is a variable for ‘consonant’, V is a variable for ‘vowel’, VV represents a long vowel, and items in parentheses are optional.

Afaan Oromo is a pitch-accent language rather than a pure tone language. That is to say, pitch is not contrastive as in tone languages and is dependent on the placement of word stress and to some degree, syntactic structure. Only the penultimate or final syllable of a root is accentuated, in which case it is realized with high pitch. In nominal expressions, the final syllable must be accented if the penultimate syllable also bears a pitch-accent. However, if the penultimate syllable does not receive an accent, then one is placed on the final syllable. With respect to verbs, accent is limited to jussive forms, which always bear an accent on their first syllable. Reduplication is productive in the language and is used primarily for plural formation and to denote distribution or intensity in nouns. Some words tend to metathesize¹. The Oromo lexicon contains numerous borrowings, primarily from European languages, but also from Arabic and Swahili.

Additionally, characters sound the same in every word in contrast to English in which the same letter may sound differently in different words. Unlike English or other Latin based languages there are no skipped or unpronounced sounds/alphabets in the language. Every alphabet is to be

¹The word completely reverse the linear order of their constituent phonemes

pronounced in a clear short/quick or long /stretched sounds. In a word where consonant is doubled the sounds are more emphasized. Besides, in a word where the vowels are doubled the sounds are stretched or elongated.

In general, Afaan Oromo has 37 basic sounds 10 vowels and 27 consonants. One possibility is to invent 37 signs corresponding to each of these 37 sounds, an impractical and unnecessary effort. Instead, it was decided that the Latin alphabet be adopted. Gemination of consonants is a very common behavior of the language. Geminates are characterized phonetically by increased duration. The phoneme sounds produced can be characterized by three things: - The manner of articulation, the place of articulation and voice of sound.

Place of articulation is about where specifically the phoneme is produced. It deals with which active articulator is making the articulation, and what part of the upper vocal tract is involved. The most common places of articulation include bilabial, labio-dental, dental, alveolar, palato-alveolar, palatal, velar, and glottal. A phoneme can be either voiced or voiceless. When voiced phonemes are produced, the vocal cord vibrates for a while. Voiceless phonemes, however, are produced without vibration of the larynx. Manner of articulation studies the way in which the consonant is produced. The articulators may close off the oral tract, or may narrow the space considerably. Based on the manner of articulation, a sound could be stops (plosives), fricatives, affricates, nasals, flap/tap, liquids, or glides [23].

Native Consonant Phonemes

A stop or plosive is a consonant sound produced by stopping the airflow in the vocal tract. Air flow is blocked to build pressure and then released. Examples are shown in Table 3.2.

Point of articulation	Voiced	Voiceless	English	Oromo
Bilabial	B	P	Bat/pat	banaa/poolisii
Alveolar	D	T	Dug/tug	Dugda/taaba
Velar	G	K	Pig/pick	Baga/bakka

Table 3. 2 Plosive Phonemes with Examples

Some consonants are produced when air is forced through a narrow opening. The narrow opening is made when two articulators come close together. A good example is the lower lip against the upper teeth to produce phoneme f. These phonemes are known as fricatives. Fricatives involve a slightly resisted flow of air. Fricatives are shown in Table 3.3.

Point of articulation	Voiced	Voiceless	English	Oromo
Labio-dental	V	f	vat/fat	vaatii/foon
Alveolar	Z	s	peas/ peace	Zayitii/saree
Palatal	ʒ	sh	measure/ mesh	-/meeshaa
Glottal		h	hat	Harree

Table 3. 3 Fricative Phonemes with Examples

Affricatives are produced when a stop and a fricative are pronounced together. Affricates are sounds that involve a plosive followed immediately by a fricative at the same location. They usually begin as stop and released as fricative as depicted in Table 3.4.

Point of articulation	Voiced	Voiceless	English	Oromo
Palate and tooth-ridge	J	ch	Judge/church	Jajuu/fiigicha

Table 3. 4 Affricate Phonemes with Examples

In nasals, the air exits through the nose rather than the mouth. They are produced with a lowered velum in the mouth, allowing air to escape freely through the nose. The oral cavity still acts as a resonance chamber for the sound, but the air does not escape through the mouth as it is blocked by the lips or tongue. All nasals are voiced [23]. Example nasal phonemes are given in Table 3.5.

Point of articulation	Voiced	Voiceless	English	Oromo
bilabial	M	-	mine	mana
alveolar	N	-	nine	nama

Table 3. 5 Nasal Phonemes with Examples

Liquids are sounds with very little air resistance. Few liquid phonemes are depicted in Table 3.6.

Point of articulation	Voiced	Voiceless	English	Oromo
Alveolar	l	-	lot	loon
Alveolar	r	-	rot	raatuu

Table 3. 6 Liquid Phonemes with Examples

Glides are similar to vowels in that there is no real contact between vocal organs. Example glides are shown in table 3.7.

Point of articulation	Voiced	Voiceless	English	Oromo
Bilabial	w	-	win	waan
Palatal	j	-	you	yoom

Table 3. 7 Glide Phonemes with Examples

Loan Consonant Phonemes

With the loaning of new words from other languages like Amharic, English, Arabic, etc. new phonemes were introduced into Afaan Oromo. Words like police, pasta, television, zeyit(zejit), tsegaye and terabeza(t'erabeza) are some of the words taken from other languages that use non-native phonemes. In some cases, these phonemes are substituted by closely similar native phonemes. The most common loan phonemes are depicted in Table 3.8.

manner	voicing	Place of articulation					
		Labial	Dental	Alveolar	Palatal	Velar	Glottal
stops	voiced						
	voiceless	p					
fricatives	voiced		v	z			
	voiceless						
	glottalized			ts [s']			

Table 3. 8 Loan Phonemes

Generally, the phonemes and articulation of Afaan Oromo consonant is summarized in Table 3.9.

Consonants (dubbifamaa)						
		Bilabial (Labiodental)	Alveolar (Retroflex)	Palato- alveolar (palatal)	Velar	Glottal
Stops and Fricatives	Voiceless	(p)	T	ch /tʃ/	K	' /ʔ/
	Voiced	B	D	j /dʒ/	g	
	Ejective	ph /pʰ/	x /tʰ/	c /tʃʰ/	q /kʰ/	
	Implosive		dh /d/			
Fricatives	Voiceless	F	S	sh /ʃ/		H
	Voiced	(V)	(z)			
Nasal		M	N	ny /ɲ/		
Approximants		W	I	y /j/		
Flap/Trill			R			

Table 3. 9 Afaan Oromo Consonants (Extracted From: IPA)

Vowel Phonemes

Afaan Oromo vowels are represented by the five basic letters such as A, E, I, O, U. Besides, it has the typical Eastern Cushitic set of five short and five long vowels by doubling the five vowel letters: ‘AA’, ‘EE’, ‘II’, ‘OO’, ‘UU’.

To differentiate between the long and short sound, however, long vowel sounds are represented by two vowels whereas short sounds are represented by single vowel. Vowel sounds, unlike consonant sounds, are produced by open oral cavity without contact between articulators. The position of the tongue and lips matter most to produce vowel sounds. Since air flow is unobstructed, all vowel sounds are voiced. They are produced by high or low position of the tongue in the mouth, and the purse of the lips [23].

Articulation of vowels describes vowel positions as the following: the front vowels which are produced at the front of the tongue, back vowels which are produced at the back of the tongue and

the central vowels which are produced in the middle of the tongue. There are also tongue heights of vowel sounds which are divided as high or close, mid, and low or open. Mid can be separated into high-mid/close-mid and low-mid/open-mid. regarding lips position, it can be rounded or unrounded vowels. The sounds o and u are rounded vowels. All the other vowel sounds are unrounded [23]. Table 3.10 shows place of articulation of vowels and tongue heights

	Place of articulation					
	Front		Center		Back	
	Gabaabaa	Dheeraa	Gabaabaa	Dheeraa	Gabaabaa	Dheeraa
Close	i	ii [i:]			u	uu [u:]
Mid	e	ee [e:]			o	oo [o:]
Open			a	aa [a:]		

Table 3. 10 Afaan Oromo Vowels

3.4 Afaan Oromo Writing System

Every language has its own rules and syntax. In English, the writing system consists of two types of letters which have different purpose. Consonants comprise most of the letters of English. Vowels are few in number and they help to produce different sounds by combining with consonants. When a vowel is added to a consonant, different variant of that consonant is produced. This helps to reduce the number of letters needed to write English, a good example why it has very few letters.

The major difference between English and Afaan Oromo language in terms of writing is how a word is spelled. In English, there is no rule how to spell or pronounce a given word and you have to memorize it by heart. This makes English difficult to write and pronounce. However, this is not a problem in Afaan Oromo writing. There are rules which anybody can follow to write as well as read words in Afaan Oromo which called (**‘Seer-luga’** in Afaan Oromo). Using these rules, a person can be sure if he/she has spelt a word correctly or not all by himself without the need of dictionary or a person good at spelling.

The writing system of the language is straightforward which is designed based on the Latin script. Unlike English or other Latin based languages there are no skipped or unpronounced

sounds/alphabets in the language. Every alphabet is to be pronounced in a clear short/quick or long /stretched sounds. In a word where consonant is doubled the sounds are more emphasized. Besides, in a word where the vowels are doubled the sounds are stretched or elongated [36].

3.4.1 Spelling Rules

The ‘**QUBEE**’ replaced the various other transliteration schemes of Afaan Oromo to Latin script and helped to standardize spelling of words. Spelling differences still occur, however, due to personal preferences and dialectal differences. Regardless, certain spelling rules can be observed that match speech patterns. Spelling mistakes in Afaan Oromo makes the misspelled word completely a different one with different meaning.

To write Afaan Oromo words there are four rules depending on the pronunciation: - ‘**Sagalee Laafaa**’ or ‘**Sagalee Jabaa**’ and ‘**Sagalee Dheeraa**’ or ‘**Sagalee Gabaabaa**’. ‘Laafaa’ and ‘Jabaa’ are differentiated by repeated consonants. When consonants of the same type are written next to each other, that sound is stressed or geminated which called (‘**Sagalee Jabaa**’, in Afaan Oromo). If only single consonant is used, it is ‘**Sagalee Laafaa**’. ‘Dheeraa’ and ‘Gabaabaa’ are deals with repeated vowels. When vowels are repeated, sounds are stretched or elongated which called ‘**Sagalee dheeraa**’. Otherwise; the sound is short which called ‘**Sagalee Gabaabaa**’.

Table 3.11 shows same example of Afaan Oromo elongated and geminated words with English meaning [23].

Laafaa	Jabaa	Meaning	Gabaabaa	Dheeraa	Meaning
Gubaa	Gubbaa	Burn/on top	Lafa	Laafaa	ground/wet
Badaa	Baddaa	Bad/highland	Fixe	Fiixee	Complish/tip
Dhibaa	Dhibba	Difficult/hundred	Bona	Boonaa	Winter/proud man

Table 3. 11 Jabaa and Laafaa, and Gabaabaa and Dheeraa

However, there are exceptions on the rule. For example, regarding digraphs or ‘**Qubee Dachaa**’ Even if it is normal that phonemes represented by digraphs are stressed like any other phonemes, in writing, they are not repeated to indicate the stress. The meaning of such words is guessed from

the sentence contextually. Hence, the same word could mean different things based on how it is used in the speech. The other exception of the rule is letter h in Afaan Oromo is not geminated.

No more than two consonants can come together without a vowel in between except when one of the consonants is not a digraph. When two different non-digraph consonants came consecutively, it is called “**Irrabuta**”, a phoneme that is uttered for very short time. Example, harka, marga, mirga, etc. in this words there is a phone [I] in between har + ka(hand), mar + ga(grass) and mir + ga(right) which uttered for short time. But, not written.

In Afaan Oromo a word cannot begin or end with a double consonant. The word for “sport” is converted to isportii. Three consonants cannot occur in a row in a word. For this reason, certain suffixes may add an i to prevent this, as in arg (“see”) + na (1st per. plu. suffix) → argina (“we see”).

Vowels cannot change without a break, either a consonant or diacritical marker called “**Hudhaa**”, between them. What breaks are used can differ with spelling preferences and dialects. For example, “very” can be **baa'ee**, **baayee**, **baa'yee**, or **baay'ee**, and “to hear” can be **dhaga'uu** or **dhagahuu**. The diacritical marker indicates that the vowels are produced independently and not as a diphthong.

3.4.2 Grammar

According to [37] the grammatical system of Afaan Oromo is quite complex and exhibits many features common to other Cushitic languages, i.e., it is an inflected language that uses postpositions more than prepositions.

Nouns and adjectives

Afaan Oromo nouns are marked for a number of categories.

- Most Afaan Oromo nouns and adjectives are marked for masculine or feminine gender.
- Nouns have an inherent masculine or feminine gender which cannot be determined by the form of the noun, with a few exceptions when biological gender is associated with a particular suffix, such as -eessa for masculine and -eetti for feminine nouns, e.g., obboleessa ‘brother’ and obboleetti ‘sister.’

- Adjectives agree with the nouns they modify in gender.
- All nouns and adjectives are marked for number: singular and plural, e.g., for masculine nouns nama ‘man’ – namicha ‘the man’; for feminine nouns haroo ‘lake’ – harittii ‘the lake’.
- All nouns are marked for case. There is a nominative, genitive, dative, accusative, instrumental, ablative, and locative case
- Nouns can be used attributively to express modification

Pronouns

Afaan Oromo pronouns have the following features:

- person: 1st, 2nd, 3rd;
- number: singular and plural;
- case: nominative, genitive, dative, accusative, instrumental, ablative, locative;
- 2nd person plural can also be used as a polite form of address;
- There is a gender distinction in the 3rd person singular but not in the plural.
- There is a distinction between proximal and distal demonstrative pronouns, e.g., kana ‘this’ and san ‘that’.

Verbs

Afaan Oromo verbs consist of a stem plus suffixes representing person, gender, number, tense-aspect, mood, and voice. Verbs agree with their subjects in person and number. Verbs, with the exception of the verb ‘be’, agree with their subjects in gender, when the subject is a 3rd person singular pronoun ‘he’ or ‘she’. There are basically two tense/aspect divisions: complete (perfective/past) and incomplete (present or future progressive). Compound tenses are formed with auxiliary verbs. There are four moods: indicative, interrogative, imperative, and jussive. The latter is used to express commands, permission, and agreement. There are three voices: active, passive, and the so-called “autobenefactive” (semi-passive/middle).

Gender

Afaan Oromo has two grammatical genders, masculine and feminine in similar way to other Afro-Asiatic language. There is no neutral gender as in other language like English [38].

Grammatical gender in Afaan Oromo enters into the grammar in the following ways:

- Verbs (except for the copula be) agree with their subjects in gender when the subject is third person singular (he or she).
- Third person singular personal pronouns (he, she, it, etc., in English) have the gender of the noun they refer to.
- Adjectives agree with the nouns they modify in gender.
- Some possessive adjectives ("my", "your") agree with the nouns they modify in some dialects.

Small number of nouns ends with -eessa(m.) and -eettii(f.), as do adjectives when they are used as nouns: Obboleessa 'brother', Obboleettii 'sister', Dureessa 'the rich one (male.)', Hiyyeettii 'the poor one (female)'. Grammatical gender normally agrees with biological gender for people and animals; thus nouns such as Abbaa 'father', Ilma 'son', and Sangaa 'ox' are masculine, while nouns such as Haadha 'mother' and Intala 'girl, daughter' are feminine.

However, most names for animals do not specify biological gender [38]. Names of astronomical bodies are feminine: Aduu 'sun', Urjii 'star'. The gender of other inanimate nouns varies somewhat among dialects.

Number

There are singular and plural numbers in Afaan Oromo [38]. But nouns that refer to multiple entities are not obligatorily plural. Some singular noun may refer multiple entities: “nama” “man” or “people”, “nama shan” “five men” or “five people”.

When it is important to make the plurality of a referent clear, the plural form of a noun is used. Noun plurals are formed through the addition of suffixes. The most common plural suffix is -ootaa;

a final vowel is dropped before the suffix, and in the western dialects, the suffix becomes -ota following a syllable with a long vowel: Mana 'house', Manoota 'houses', Hiriya'a 'friend', Hiriyoota 'friends', Barsiisaa 'teacher', Barsiiso(o)ta 'teachers'. Among the other common plural suffixes are -(w)wan, -een, and -(a)an; the latter two may cause a preceding consonant to be doubled: waggaa 'year', waggaa'wan 'years', laga 'river', laggeen 'rivers', ilma 'son', ilmaan 'sons'.

Definiteness

There is no indefinite article (such as a, an, some) in Afaan Oromo like they exist in English. The definiteness article 'the' in English is (t)icha for masculine nouns (the **ch** is geminated though this is not normally indicated in writing) and -(t)ittii for feminine nouns in Afaan Oromo. Vowel endings of nouns are dropped before these suffixes: karaa 'road', karicha 'the road', nama 'man', namicha/namticha 'the man', haroo 'lake', harittii 'the lake'. Note that for animate nouns that can take either gender, the definite suffix may indicate the intended gender: qaalluu 'priest', qaallicha 'the priest (male)', qallittii 'the priest (female)'. The definite suffixes appear to be used less often than the in English, and they seem not to co-occur with the plural suffixes [38].

Afaan Oromo Nouns

Afaan Oromo nouns are presented in their respective examples of each type so as to simplify for the nouns used in the language under investigation.

Nominative

The nominative is used for nouns that are the subjects of clauses.

- *Ibsaa* man's name, *Ibsaan* 'Ibsaa (nom.)', *konkoolaataa*, *qaba* 'he has', *Ibsaan makiinaa* *qaba* 'Ibsaa has a car'

Genitive

The genitive is used for possession or "belonging"; it corresponds roughly to English *of* or *'s*. The genitive is usually formed by lengthening a final short vowel, by adding *-ii* to a final consonant,

and by leaving a final long vowel unchanged. The possessor noun follows the possessed noun in a genitive phrase. Many such phrases with specific technical meanings have been added to the Oromo lexicon in recent years.

- *obboleetti* 'sister', *namicha* 'the man', *obboleetti namichaa* 'the man's sister'
- *hojii* 'job', *Caaltuu*, woman's name, *hojii Caaltuu*, 'Caaltuu's job'
- *barumsa* 'field of study', *afaan* 'mouth, language', *barumsa afaanii* 'linguistics'

In place of the genitive it is also possible to use the relative marker *kan* (m.) / *tan* (f.) preceding the possessor.

- *obboleetti kan namicha* 'the man's sister'

Dative

The dative is used for nouns that represent the recipient (*to*) or the benefactor (*for*) of an event. The dative form of a verb infinitive (which acts like a noun in Oromo) indicates purpose. The dative takes one of the following forms:

- Lengthening of a final short vowel (ambiguously also signifying the genitive)
- *namicha* 'the man', *namichaa* 'to the man, of the man'

Instrumental

The instrumental is used for nouns that represent the instrument ("with"), the means ("by"), the agent ("by"), the reason, or the time of an event. The formation of the instrumental parallels that of the dative to some extent:

- *-n* following a long vowel or a lengthened short vowel; *-iin* following a consonant
- *harka* 'hand', *harkaan* 'by hand, with a hand'
- *Afaan Oromoo* 'Oromo (language)', *Afaan Oromootiin* 'in Oromo'

Locative

The locative is used for nouns that represent general locations of events or states, roughly *at*. For more specific locations, Oromo uses prepositions or postpositions. Postpositions may also take the locative suffix. The locative also seems to overlap somewhat with the instrumental, sometimes having a temporal function. The locative is formed with the suffix *-tti*.

- *guyyaa* 'day', *guyyaatti* 'per day'
- *jala*, *jalatti* 'under'

Ablative

The ablative is to represent the source of an event; it corresponds closely to English *from*. The ablative, applied to postpositions and locative adverbs as well as nouns proper, is formed in the following ways: *keessa* 'inside, in', *keessaa* 'from inside'

- *finfinnee* 'finfine(Addis Ababa)', *Finfinneedhaa* 'from Finfinnee (Addis Ababa)'
- *gabaa* 'market', *gabaadhaa* 'from market'

Word order

The typical word order in Oromo sentences is Subject – Object – Verb (SOV). Modifiers, articles, pronouns, and case markers follow the nouns they modify. The words phrases and simple sentence selected for this researcher are selected and collected considering all the above characteristics of the Afaan Oromo language and the phonemes. This helped the researcher in finding the characteristics of the language and representing the language for better results of the thesis. Furthermore, the expert in the domain (linguistics and phonetics) were consulted from the linguistics and phonetics of Afaan Oromo.

CHAPTER 4

CMU SPHINX: THE RECOGNIZER LIBRARY

In this chapter we will try to discuss the tools that are used to build the entire system. The tools that are used for Acoustic model building, language model building and the decoder for speech recognition results. The CMU Sphinx toolkit is a speech recognition tool kit with various tools used to build speech applications; it has a number of packages for different tasks and applications.

4.1 Why Sphinx

As a developer, the major reason to use Sphinx is because Sphinx currently is the most complete archive which is in Berkeley's license. We are aware of that other very powerful toolkits, are also open-sourced. Unfortunately, some of their licenses disallow to use their code in any project. It may also disallow to use the code and distribute them later in the project.

Another aspect of Sphinx is that based on Sphinx and its related resources, and with enough knowledge, allows building a lot of different types of applications. By speech applications, mean things like dictation, automatic speech server. More importantly, the most common denominator or all these applications, i.e the recognizer is open-sourced in CMU Sphinx. It also gives a lot of bundling resource such as dictionary, phone list from CMU's speech archive. This is something very important if you want to have full-control of speech recognition system.

As a researcher and if the interest is in building HMM-based speech recognition, there are several reasons why we want to use Sphinx 3 and Sphinx 4 as our research tools.

First of all Sphinx assumes that GMM computation and the search process are separate. The important consequence is that we could carry out two different types of research without conflicting each other's. For example, we could try to device a new observation probability without touching the source code of the search routine. At the same time, we could also build a new search algorithm without thinking of the GMM computation. This gives us a better advantage in speech recognition research.

The second thing is in training. Most of the time when we were doing modeling research, what we would like to change is the algorithm of estimation. SphinxTrain's Baum-Welch's algorithm solves this problem in two stages. It first dumps the posterior probabilities count in a separate file and can be easily readable by libraries of SphinxTrain. This advantage can greatly shorten our research time from weeks to dates.

The third reason is in the code-clarity; later recognizers such as Sphinx 3 and Sphinx 4 have put readability of the code as one of the most important design goal. Many researchers, not only they want to use the recognizer as a tool. They also want to change the code to suit their purpose. Sphinx 3, for example, has only 4 layers of code from main () to the GMM computation. These become a good advantage for researchers who like to make changes in a speech recognizer.

The final reason is in speed; all Sphinx's decoders are very fast because the speed of the recognizer is always one important concern. We will not only able to build a research system easily but quickly. This is usually an important factor why we would like to use Sphinx.

4.2 Acoustic Modelling Toolkit

SphinxTrain: A good decoder cannot live without a good trainer. SphinxTrain's trainer has an equally interesting development history as in the decoder.

At the age when Sphinx-I and Sphinx-II is created and widely used internally CMU. Experiments of acoustic modeling and language modeling are usually done by "researcher's code". That means individual researchers hack out a tools using C or perl and full fill the aim of training. The obvious problem is data formats might not be exchangeable and caused a lot of problems in advancing speech recognition research.

4.3 Language Modelling Tool kit

SRILM: SRILM is a toolkit for building and applying statistical language models (LMs), primarily for use in speech recognition, statistical tagging and segmentation, and machine translation. It has been under development in the SRI Speech Technology and Research Laboratory since 1995.

4.4 Decoding Toolkit

Sphinx-4 is a flexible, modular and pluggable framework to help foster new innovations in the core research of hidden Markov model (HMM) recognition systems. The design of Sphinx-4 is based on patterns that have emerged from the design of past systems as well as new requirements based on areas that researchers currently want to explore. To exercise this framework, and to provide researchers with a “research-ready” system, Sphinx-4 also includes several implementations of both simple and state-of-the-art techniques [39].

The Sphinx-4 framework has been designed with a high degree of flexibility and modularity. Figure 4.1 shows the overall architecture of the system. Each labeled element in Figure 5.1 represents a module that can be easily replaced, allowing researchers to experiment with different module implementations without needing to modify other portions of the system.

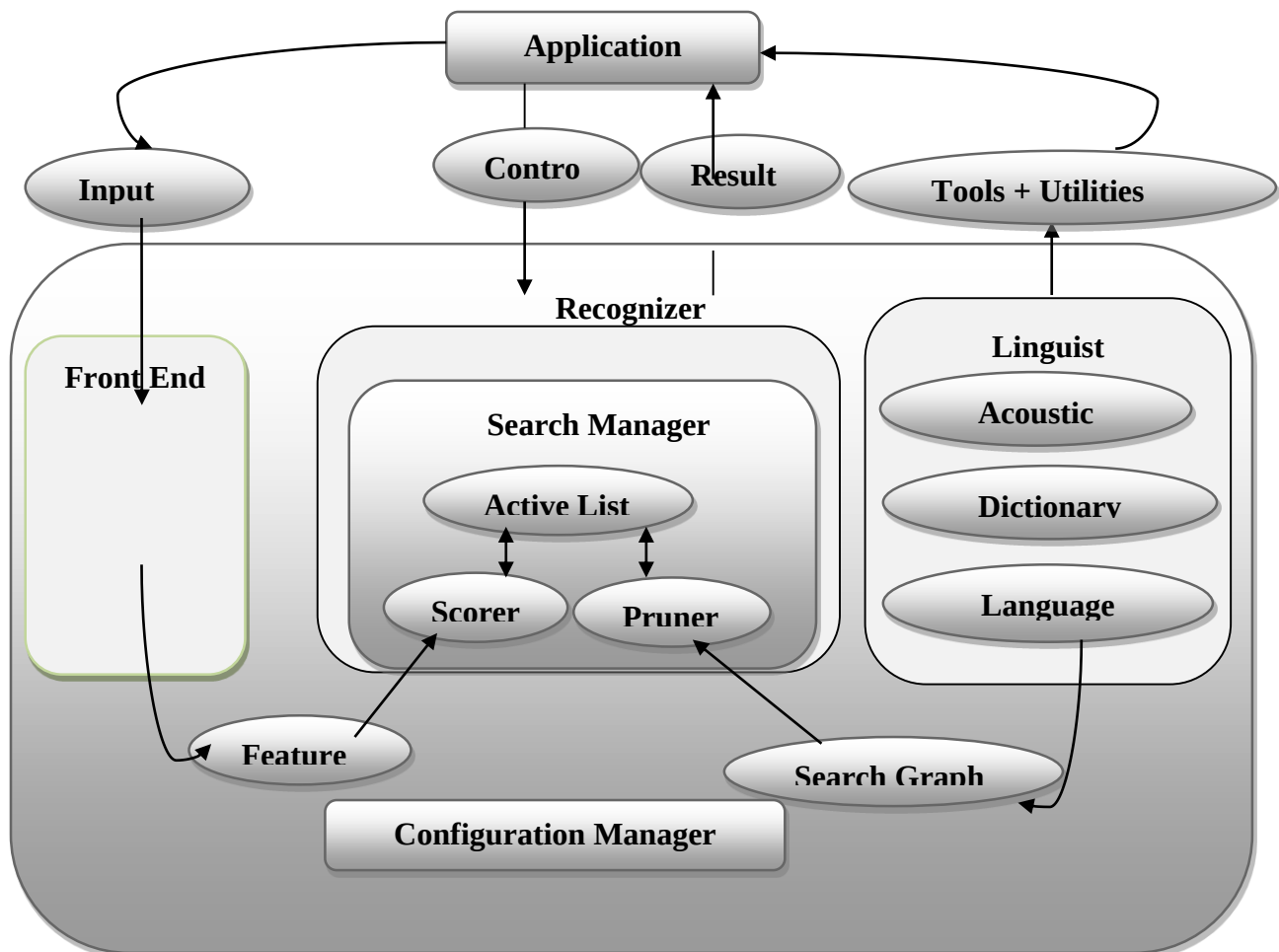


Figure 4 1 Sphinx-4 Decoder Framework. The main blocks are the Front End, the Decoder, and the Linguist. Adapted from [39]

There are three primary modules in the Sphinx-4 framework: the *Front End*, the *Decoder*, and the *Linguist*. The Front End takes one or more input signals and parameterizes them into a sequence of Features. The Linguist translates any type of standard language model, along with pronunciation information from the Dictionary and structural information from one or more sets of Acoustic Models, into a Search Graph. The Search Manager in the Decoder uses the Features from the Front End and the Search Graph from the Linguist to perform the actual decoding, generating Results. At any time prior to or during the recognition process, the application can issue Controls to each of the modules, effectively becoming a partner in the recognition process.

A. Front End

The purpose of the Front End is to parameterize an Input signal (e.g., audio) into a sequence of output Features. As illustrated in Figure 4.2, the Front End comprises one or more parallel chains of replaceable communicating signal processing modules called Data Processors. Supporting multiple chains permits simultaneous computation of different types of parameters from the same or different input signals. This enables the creation of systems that can simultaneously decode using different parameter types, such as MFCC and PLP, and even parameter types derived from non-speech signals such as video [39].

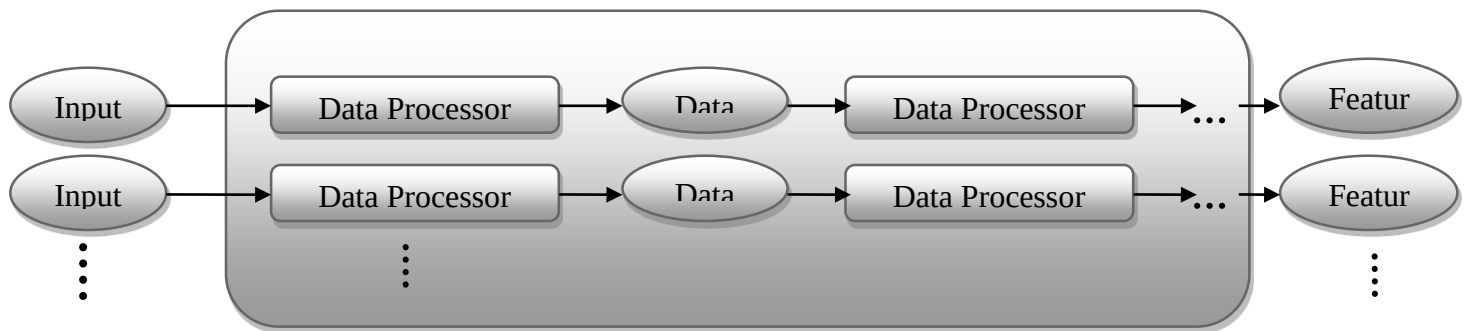


Figure 4 2 Sphinx-4 Front End. The Front End comprises one or more parallel chains of communicating Data Processors Adapted from [39]

B. Linguist

The Linguist generates the Search Graph that is used by the decoder during the search, while at the same time hiding the complexities involved in generating this graph. As is the case throughout

Sphinx-4, the Linguist is a pluggable module, allowing people to dynamically configure the system with different Linguist implementations.

A typical Linguist implementation constructs the Search Graph using the language structure as represented by a given Language Model and the topological structure of the Acoustic Model (HMMs for the basic sound units used by the system). The Linguist may also use a Dictionary (typically a pronunciation lexicon) to map words from the Language Model into sequences of Acoustic Model elements. When generating the Search Graph, the Linguist may also incorporate sub-word units with contexts of arbitrary length, if provided.

By allowing different implementations of the Linguist to be plugged in at runtime, Sphinx-4 permits individuals to provide different configurations for different system and recognition requirements. For instance, a simple numerical digits recognition application might use a simple Linguist that keeps the search space entirely in memory. On the other hand, a dictation application with a 100K word vocabulary might use a sophisticated Linguist that keeps only a small portion of the potential search space in memory at a time.

The Linguist itself consists of three pluggable components: the Language Model, the Dictionary, and the Acoustic Model, which are described in the following sections.

Language Model

The Language Model module of the Linguist provides word-level language structure, which can be represented by any number of pluggable implementations. These implementations typically fall into one of two categories: graph-driven grammars and stochastic N-Gram models. The graph-driven grammar represents a directed word graph where each node represents a single word and each arc represents the probability of a word transition taking place. The stochastic N-Gram models provide probabilities for words given the observation of the previous $n-1$ words.

The Sphinx-4 Language Model implementations support a variety of formats, including the following:

- *Simple Word List Grammar*: defines a grammar based upon a list of words. An optional parameter defines whether the grammar “loops” or not. If the grammar does not loop, then the grammar will

be used for isolated word recognition. If the grammar loops, then it will be used to support trivial connected word recognition that is the equivalent of a unigram grammar with equal probabilities.

- *JSGF Grammar*: supports the Java TM Speech API Grammar Format (JSGF), which defines a BNF-style, platform independent and vendor-independent Unicode representation of grammars.
- *LM Grammar*: defines a grammar based upon a statistical language model. LM Grammar generates one grammar node per word and works well with smaller unigram and bigram grammars of up to approximately 1000 words.
- *FST Grammar*: supports a finite-state transducer (FST) in the ARPA FST grammar format.
- *Simple N-Gram Model*: provides support for ASCII N-Gram models in the ARPA format. The Simple N-Gram Model makes no attempt to optimize memory usage, so it works best with small language models.
- *Large Trigram Model*: provides support for true N-Gram models generated by the CMU-Cambridge Statistical Language Modeling Toolkit. The Large Trigram Model optimizes memory storage, allowing it to work with very large files of 100MB or more.

Dictionary

The Dictionary provides pronunciations for words found in the Language Model. The pronunciations break words into sequences of sub-word units found in the Acoustic Model. The Dictionary interface also supports the classification of words and allows for a single word to be in multiple classes.

Sphinx-4 currently provides implementations of the Dictionary interface to support the CMU Pronouncing Dictionary. The various implementations optimize for usage patterns based on the size of the active vocabulary. For example, one implementation will load the entire vocabulary at system initialization time, whereas another implementation will only obtain pronunciations on demand.

Acoustic Model

The Acoustic Model module provides a mapping between a unit of speech and an HMM that can be scored against incoming features provided by the Front End. As with other systems, the

mapping may also take contextual and word position information into account. For example, in the case of tri-phones, the context represents the single phonemes to the left and right of the given phoneme, and the word position represents whether the tri-phone is at the beginning, middle, or end of a word (or is a word itself). The contextual definition is not fixed by Sphinx-4, allowing for the definition of Acoustic Models that contain allophones as well as Acoustic Models whose contexts do not need to be adjacent to the unit.

Typically, the Linguist breaks each word in the active vocabulary into a sequence of context-dependent sub-word units. The Linguist then passes the units and their contexts to the Acoustic Model, retrieving the HMM graphs associated with those units.

It then uses these HMM graphs in conjunction with the Language Model to construct the Search Graph.

Unlike most speech recognition systems, which represent the HMM graphs as a fixed structure in memory, the Sphinx-4 HMM is merely a directed graph of objects. In this graph, each node corresponds to an HMM state and each arc represents the probability of transitioning from one state to another in the HMM. By representing the HMM as a directed graph of objects instead of a fixed structure, an implementation of the Acoustic Model can easily supply HMMs with different topologies. For example, the Acoustic Model interfaces do not restrict the HMMs in terms of the number of states, the number or transitions out of any state, or the direction of a transition (forward or backward). Furthermore, Sphinx-4 allows the number of states in an HMM to vary from one unit to another in the same Acoustic Model.

Each HMM state is capable of producing a score from an observed feature. The actual code for computing the score is done by the HMM state itself, thus hiding its implementation from the rest of the system, even permitting differing probability density functions to be used per HMM state. The Acoustic Model also allows sharing of various components at all levels. That is, the components that make up a particular HMM state such as Gaussian mixtures, transition matrices, and mixture weights can be shared by any of the HMM states to a very fine degree.

As with the rest of Sphinx-4, individuals can configure Sphinx-4 with different implementations of the Acoustic Model based upon their needs. Sphinx-4 currently provides a single Acoustic Model implementation that is capable of loading and using acoustic models generated by the Sphinx-3 trainer.

C. Decoder

The primary role of the Sphinx-4 Decoder block is to use Features from the Front End in conjunction with the Search Graph from the Linguist to generate Result hypotheses. The Decoder block comprises a pluggable Search Manager and other supporting code that simplifies the decoding process for an application. As such, the most interesting component of the Decoder block is the Search Manager.

The Decoder merely tells the Search Manager to recognize a set of Feature frames. At each step of the process, the Search Manager creates a Result object that contains all the paths that have reached a final non-emitting state. To process the result, Sphinx-4 also provides utilities capable of producing a lattice and confidence scores from the Result. Unlike other systems, however, applications can modify the search space and the Result object in between steps, permitting the application to become a partner in the recognition process.

Like the Linguist, the Search Manager is not restricted to any particular implementation. For example, implementations of the Search Manager may perform search algorithms such as frame-synchronous Viterbi, A*, bi-directional, and so on.

Each Search Manager implementation uses a token passing algorithm as described by Young [24]. A Sphinx-4 token is an object that is associated with a Search State and contains the overall acoustic and language scores of the path at a given point, a reference to the Search State, a reference to an input Feature frame, and other relevant information. The Search State reference allows the Search Manager to relate a token to its state output distribution, context-dependent phonetic unit, pronunciation, word, and grammar state. Every partial hypothesis terminates in an active token.

CHAPTER 5

EXPERIMENTATION AND ANALYSIS

5.1 System Architecture

The adapted recognizer design used in the entire development of the prototype for Afaan Oromo Automatic Speech Recognition is given in Figure 5.1.

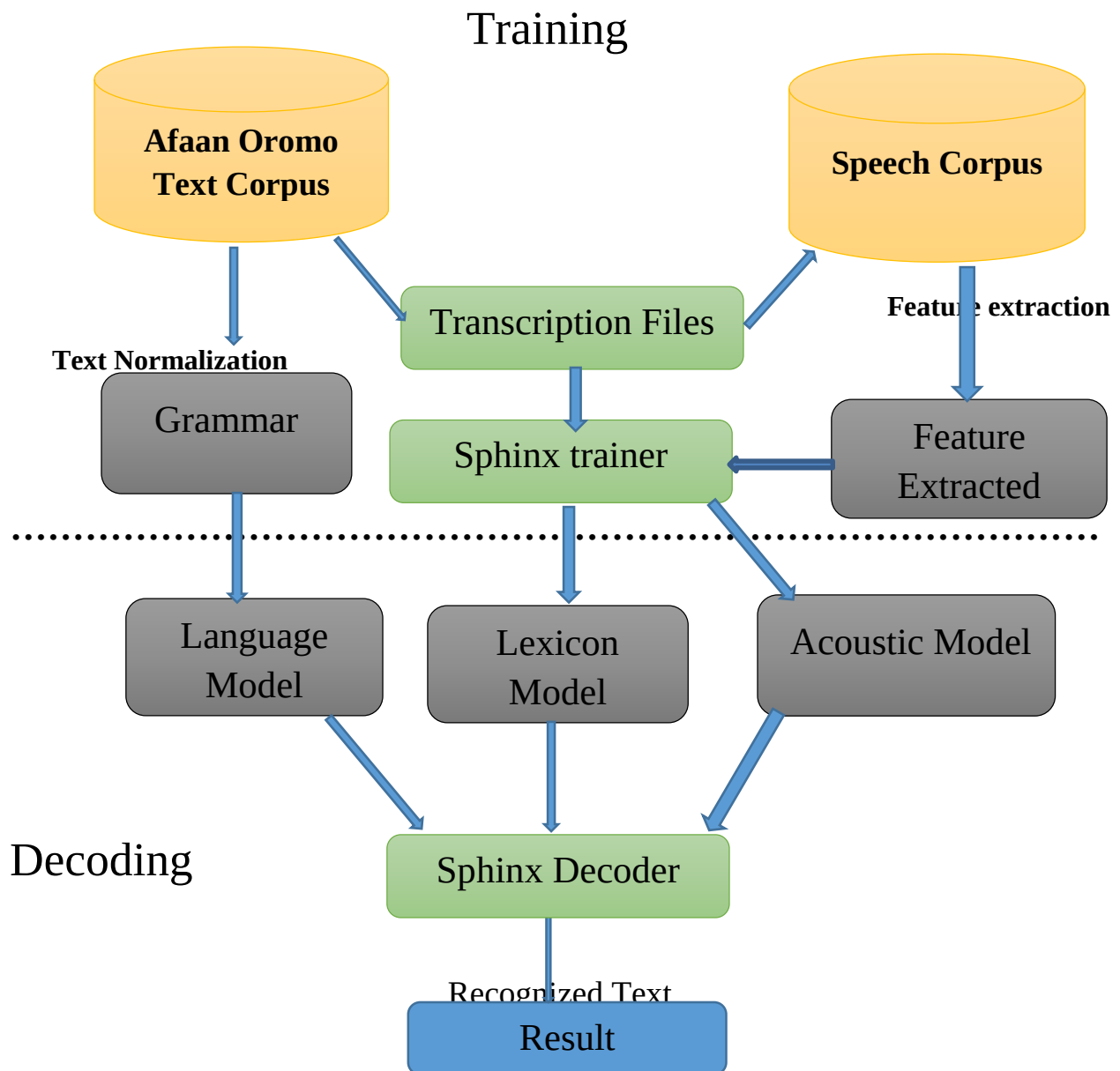


Figure 5. 1 Afaan Oromo ASR System Architecture Using CMU Sphinx

5.2 Data Preparation

Data preparation is the first task that should be addressed in speech recognition development process. The speech corpus is designed according to best-practice guidelines established for other languages. Standard speech corpora consist of a training set and evaluation test sets. The training set is intended to collect speech data for training the recognizer and the evaluation test set is for the purpose of final evaluation of the recognizer.

The selection of sentences aims at both a phonetically rich and balanced collection of sentences. To accomplish phonetic richness, we select 700 training and 100 evaluation test sentences by consulting domain experts. The text corpus are collected from different Afaan Oromo sources (Fana Broadcasting Corporation (FBC), Voice of America (VOA), Books, Kassahun [15] and Teferi [23] transcription texts which contributed to inclusion all Afaan Oromo phones. The text corpus is shown in Appendix A.

After text selection we performed text normalization to make it ready made for speech recognition tool. These include abbreviations are expanded, symbols are removed, numbers are converted into equivalent text, punctuation marks are removed, etc.

For this study, having the text corpus 800 sentences are selected for recording in which 700 sentences for training and 100 for testing. The corpus is recorded by sixteen speakers (Eleven male and Five female) in different environment by using microphone. This is mainly because the researcher couldn't get access to readers in one place and lack of available sound recording room. Some of speech data is recorded in Adama City, Goro Secondary and Preparatory School.

The prepared sets of training and evaluation test sets are recorded using Praat software with the sample frequency of 16 kHz, sample size of 16 bits and Mono channel. The training and evaluation sentences are saved in the transcription File Format (.transcription) and Audio data are saved in Wave format. The training and evaluation test sets are recorded by speakers in the age range of 15 to 30. Here are some example of Afaan Oromo transcription file selected for training and testing what uttered by speakers.

<s> Aadaan calaqqee jiruu fi jireenya saba tokkooti </s> (tr01-001)

<s> Saba dinagdeen guddatee fi siyaasaan qaroomme </s> (tr01-002)

.....

<s> Dhaabbanni Bosonaafi Bineensota Bosona Oromiyaa bosona lafa hektaara miliyoona lama caalurra jiru daangesse </s> (tr01-050)

Transcription file (your_db_train.transcription and your_db_test.transcription) is a text file listing the transcription for each audio file.

<s> ijoollee kiyya abbaan keenya duhe jettanii hin gaddinaa </s> (ts01-001)

<s> bakka dhugaa dhaabbadhee dhugaa dubbadheen duhe </s> (ts01-002)

.....

<s> ani saawwa tiksa ture </s> (ts02-050)

It's important that each line starts with <s> and ends with </s> followed by id in parentheses. Also note that parenthesis contains only the file, without speaker_n directory. It's critical to have exact match between fileids file and the transcription file. The number of lines in both should be identical. Last part of the file id (train/**tr01-001**) and the utterance id tr01-001 must be the same on each line.

5.3 Pronunciation Dictionary

The dictionary provides pronunciations for words found in the language model, and the acoustic models which may be composed of sub word (phonetic, syllabic, etc.) units. A pronunciation dictionary is a machine-readable transcription of words in terms of sub-word units. It specifies the finite set of words that may be output by the speech recognizer and gives, at least, one pronunciation for each. In pronunciation variation research, one is usually confronted with two types of lexica: a canonical (or baseline) lexicon and a multiple pronunciation lexicon.

A canonical lexicon contains the normative or standard transcriptions for the words; this is a single transliteration per word. A multiple pronunciation lexicon contains more than one transliteration variant per word, for some or all of the words in the lexicon.

In this study the pronunciation variation does not considered because of our training data set is small. So that we developed our own canonical pronunciation dictionary for this study. In the

previous attempts of read speech Afaan Oromo recognizer development, Kassahun [15] used the pronunciation dictionary created by LM tool which provides pronunciation dictionary for small text data On-Line.

To develop our lexical model we used a total of 2024 words that are uttered in the training data. In order to generate the lexical model since there is no any common program that is used for making dictionaries we developed our own program using java programming that is used for generating pronunciation dictionary automatically. In order to generate this lexical model we used 56 phones of Afaan Oromo. We developed a canonical dictionary see Appendix B.

5.4 Acoustic Model

Our acoustic model is developed with Sphinx Trainer. In order to train the acoustic model we used a read speech of 700 sentences uttered by 14 speakers in which every speaker utters 50 unique sentences. During the preparation of the acoustic model we have used different tools beyond Sphinx Trainer and also we have to delve how to prepare best acoustic model with this tool.

Before we have to train using our tools we followed lots of procedures. The speech data was extremely noisy and was not segmented, because of this we have to segment the speech data into sentence level speech to do that we used Praat speech processing tool for segmenting the speech. Detailed description is shown in the next sections.

5.4.1 Signal Segmentation and Annotation

The recorded speech that saved in Waveforms need to be segmented to shorter parts and annotated. The recordings have been segmented and annotated fully manually but we suppose to use some automated or semi-automated procedure for the next collection for saving of hard manual work.

Freely distributed speech segmentation, labeling, and transcription software Praat is used for signal segmentation. It gives full support for the first steps in speech corpus creation. Each utterance is divided into sentences, which are the most suitable form for the next processing and for training of ASR system. For read speech, segmentation is done based on transcription file. In case of spontaneous the transcription is not known, the start- and end-points of the sentences are found manually.

The segmented speech and transcription file should have to match each other, unless CMU sphinx trainer will generate mismatch error. During segmentation we found many mismatch between recorded speech data and prepared transcription file. We re-edited our transcription file based on how the speaker uttered. And also we removed long pauses manually. It is very tedious work.

The speech is written in the form it is exactly spoken, even in the case of colloquial language or mathematical expressions, which are typically present in technical speech. We did not use any special symbols or representations for spelling, mispronunciations and foreign words. Punctuation is not noted as it is not important for training a phoneme model-based speech recognizer. Sphinx support both Unicode language and ASCII characters for orthographic annotation text, Afaan Oromo uses ASCII.

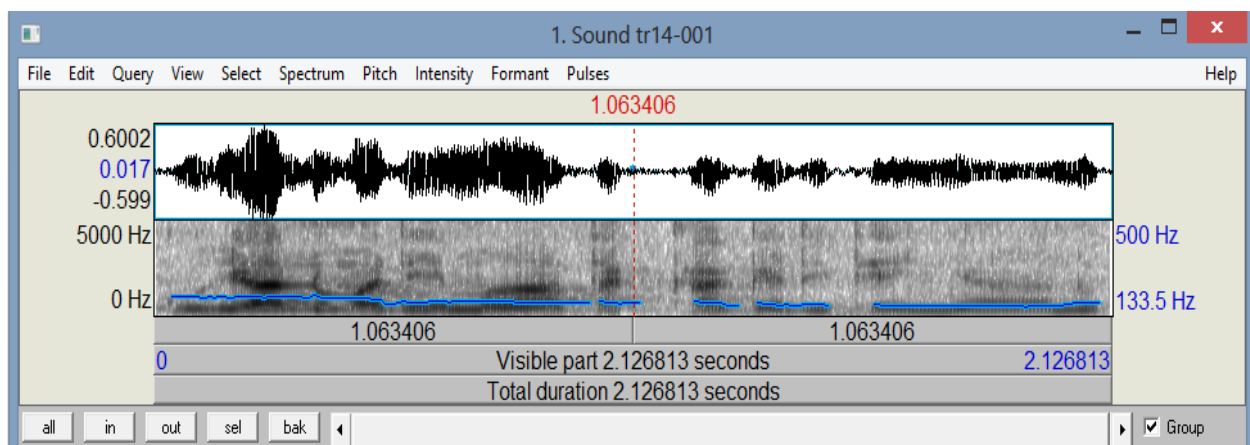


Figure 5. 2 Praat Speech Segmentation, Labeling, and Transcription Software

We used for our acoustic model the continuous model procedure because it is recommended for speech data recorded with 16 KHz sampling rate and also it is recommended for sphinx-3 decoder and above. Semi-continuous models are recommended for speech data of telephone conversations and speech that are recorded with 8 KHz sampling rate.

5.4.2 Feature Extraction

The feature extraction stage seeks to provide a compact representation of the speech waveform. This form should minimize the loss of information that discriminates between words, and provide a good match with the distributional assumptions made by the acoustic models. For example, if

diagonal covariance Gaussian distributions are used for the state-output distributions then the features should be designed to be Gaussian and uncorrelated.

Feature vectors are typically computed every 13 ms using an overlapping analysis window of around 25 ms. one of the simplest and most widely used encoding schemes is based on Mel-Frequency Cepstral coefficients (MFCCs). These are generated by applying a truncated discrete cosine transformation (DCT) to a log spectral estimate computed by smoothing an FFT with around 20 frequency bins distributed non-linearly across the speech spectrum.

The nonlinear frequency scale used is called a mel scale and it approximates the response of the human ear. The DCT is applied in order to smooth the spectral estimate and approximately decorrelate the feature elements. After the cosine transform the first element represents the average of the log-energy of the frequency bins. This is sometimes replaced by the log-energy of the frame, or removed completely. Table 5.1 show all parameters used in feature extraction.

Parameter	Value
Sampling Rate	16Khz
Pre-emphases Coefficient	0.97
Window Size	25ms
Overlap Duration	15ms
Hamming Window	True
Zero Mean	True
Cepstral lifter	22
Number of Cepstral Coefficient	13

Table 5. 1 Parameters Used in Feature Extraction

5.4.3 Acoustic Model Training Files

During training of our acoustic model we used continuous read speech corpus of 700 sentences uttered by 14 speakers, our pronunciation dictionary is canonical dictionary which has a total of 2024 unique words, and the total phone coverage we used is 56 phones which includes 55 basic Afaan Oromo phones and 1 silence phone. The SIL phoneme is added to the phone list so as to handle the silence filler dictionary that might cause errors during training.

Afaan Oromo phones are listed on Appendix C. We organized all the prepared data into two folders the etc and wav folder in which each folders holds different data.

etc

- AOASR.dic - *Phonetic dictionary*
- AOASR.phone - *Phone set file*
- AOASR.lm.bin - *Language model*
- AOASR.filler - *List of fillers*
- AOASR_train.fileids - *List of files for training*
- AOASR_train.transcription - *Transcription for training*
- AOASR_test.fileids - *List of files for testing*
- AOASR_test.transcription - *Transcription for testing*

wav

speaker1

- tr01-001.wav - *Recording of speech utterance*

speaker2

- tr02-001.wav

During feature extraction stage we got a set of feature files computed from the audio training data we collect for every recording we have in the training corpus. Each recording is transformed into a sequence of feature vectors using a front-end executable provided with SPHINX training package called sphinx_fe.

```
Perl scripts_pl/000.comp_feat/slave_feat.pl
```

These scripts used to extract feature file from acoustic signal. During feature extraction the feature files are stored in a different folder called feat. A **Control File (AOASR_train.fileids)** containing the list of feature-set filenames with full paths to them also provided for the training of acoustic model. An example of the entries in this file:

train/tr01-001

train/tr01-002

File extensions are not given for control file it provided to the trainer in the configuration file.

A **Transcript File (AOASR_train.transcription)** in which the transcripts corresponding to the feature files are listed in exactly the same order as the feature filenames in the control file.

<s> Aadaan calaqqee jiruu fi jireenya saba tokkooti </s> (tr01-001)

<s> Saba dinagdeen guddatee fi siyaasaan qarooma </s> (tr01-002)

The transcript file is also located in the same folder like the control file in which the trainer configuration file easily locate each files during training. A **Main Dictionary (AOASR.dic)** which has all acoustic events and words in the transcripts mapped onto the acoustic units.

aadaa aa d aa

aadaadhuma aa d aa dh u m a

A **Filler Dictionary (AOASR.filler)** contains filler phones (not-covered by language model non-linguistic sounds like breathe, hmm or laugh). It can contain just silences:

<s> SIL

</s> SIL

<sil> SIL

A **Phone List (AOASR.phone)**, which is a list of all acoustic units that we used for training. We used total of 56 phones including phones that are used for mapping non-speech events. It should have one phone per line. The number of phones should match the phones used in the dictionary plus the special SIL phone for silence:

SIL

a

aa

b

bb

wav directory is used to store all the audio recordings that are used for training.

During acoustic model training of our data we followed the procedure that is shown on the sphinx trainer chart.

SETTING UP THE TRAINING SCRIPTS

To start the training we have to change our path to the database folder and run the following commands using terminal in Linux environment.

```
sphinxtrain -t AOASR setup
```

This will copy all the required configuration files into etc subfolder of AOASR database folder and prepare database for training. After setup, we need to edit the configuration files in etc folder, there are many variables but we used few of them depending on our experiment. sphinx_train.cfg adjustment depends on what we need to train. Appendix D shows sphinx_train.cfg adjustment done for CD context dependent model using 8 Gaussian mixture.

5.5 Language Model

Language models (LM) are fundamental to many natural language applications such as automatic speech recognition (ASR) and statistical machine translation (SMT).

A statistical language model (SLM) is a probability distribution $P(W)$ over word sequences W that models how often each sequence W occurs as a sentence. For this study, we developed language model using SRILM tool kit. The text we used for language model development is collected from different Afaan Oromo source (Fana Broadcasting Corporation (FBC), Voice of America (VOA), Books, E-materials, etc.) like transcription texts.

After text selection we performed text normalization to make it ready made for LM toolkit. These include abbreviations are expanded, symbols are removed, numbers are converted into equivalent text, punctuation marks are removed, etc. We used 10000 Afaan Oromo normalized sentences. Since, language model has a huge effect on recognition performance we have used Additive or Laplace smoothing technique to build the best language model. We built both Bi-gram language model and tri-gram language model for this study.

5.6 Recognizer Testing and Evaluation

The heart of automatic speech recognition is the search for the most likely word sequence given the observed features extracted from the speech signal. This is commonly referred to as decoding or recognizing the speech signal. When decoding speech, we begin 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 utterance. Because the Viterbi algorithm is used to find the most likely word sequence, the decoding procedure is said to be done via Viterbi search. We have prepared both test and training data, we have built language and lexical models and we have performed training which comes up with Acoustic model in our previous activities. Now we can say that the recognizer is built and we can perform recognition and then evaluate the result.

The designed ASR model is evaluated under the Word Error Rate (WER) and Sentence Error Rate (SER) criterion. The WER metric is the ratio of the number of recognition errors to the number of words in the reference. The number of recognition errors is calculated as the minimum number of insertion, substitution or deletion operations required to obtain the same string as the reference from the recognizer output. The general difficulty of measuring performance lies in the fact that the recognized word sequence can have a different length from the reference word sequence. The error rate is therefore derived at the word level instead of the phoneme level.

$$\text{WER} = \frac{\text{Substitutions} + \text{Deletions} + \text{Insertions}}{\text{\# words in reference}} \times 100$$

Sentence Error Rate (SER) is the number of sentences with at least one word error.

$$SER = \frac{\text{\# of sentences with at least one word error}}{\text{Total number of sentences}} \times 100$$

The first task we have used for our system evaluation is preparing test data for evaluating the recognizer performance.

We prepared two test data sets, one for speaker independent the other for speaker dependent in which 100 sentences uttered by two speakers are used for both cases. Because the number of female speakers are small we used only male speaker utterance in test set. We evaluate the performance of the recognizer that is developed with sphinx-4 decoder and Pocket sphinx pre-alpha.

The evaluation is based on the acoustic model parameters used in training process and type of language model used in testing process. The language model we have used for this study were both bigram and tri-gram. In order to improve the recognizer performance we did many experiments by periodically tune the number of an acoustic model parameters and by smoothing language model. We did experiment by changing the number of Gaussian mixture and tied state (senones) and also we used Additive or Laplace language model smoothing techniques, then trained repeatedly and come up with different recognition results.

In order to improve the recognizer performance we increased the number of Gaussian mixture then trained repeatedly and come up with acoustic models with different mixtures but the recognizer performance starts decreasing at eight (8) Gaussian mixture. Therefore all the results we have reported are the results found at eighth Gaussian mixture.

The results are given below in tables and followed by discussion. The performance of the recognizer is tested for both speaker independent and dependent test data set, finally the result is reported according to the experimentation.

The experimentation is conducted in two distinct phases each phase were tested using speaker dependent and independent test data set.

Phase I Experimentation

In this phase the first experiment conducted for speaker dependent using different parameter of tied state and bi-gram language model. The acoustic model is developed with skip transition topology and 8 Gaussian mixtures. The recognition performance on the below table 5.2 shows recognition result of the experiment.

Speaker Code	Language Model	Word Accuracy (using 200 tied state)	Word Accuracy (using 500 tied state)	Word Accuracy (using 1000 tied state)
ts01	Bigram	63.4%	84.2%	93%
ts02	Bigram	56.4%	79%	90.4%
Average Result		59.9%	81.6%	91.7%

Table 5. 2 Speaker Dependent Using 8 Gaussian Mixture with Skip

As we can observe from the above table the analysis result for tied state values 200, 500, and 1000 using bigram language model is 59.9%, 81.6% and 91.7% respectively. The recognizer performs better at 1000 tied state.

Table 5.3 shows the recognition result obtained in case of trigram language model.

Speaker Code	Language Model	Word Accuracy (using 200 tied state)	Word Accuracy (using 500 tied state)	Word Accuracy (using 1000 tied state)
ts01	Tri-gram	60.4%	79.4%	89.6%
ts02	Tri-gram	53.4%	80.4%	86.8%
Average Result		56.9%	79.9%	88.2%

Table 5. 3 Speaker Dependent Using 8 Gaussian Mixture with Skip

As we can observe from the above table the analysis result for tied state values 200, 500, and 1000 using trigram language model is 56.9%, 79.9% and 88.2% respectively. The recognizer performs better at 1000 tied state.

As we can see from table 5.2 and 5.3 word accuracy 91.7% recognized for 1000 tied state using bigram language model, while we obtain 88.2% word accuracy in 1000 tied state for trigram

language model. And also as we can see from the above two tables for speaker dependent test data set, using different value of tied state the recognizer performs better for bigram language model at each tied state value comparing to trigram language model.

The second experimentation is done for speaker independent using the same parameter used for speaker dependent experiment mentioned above.

Speaker	Language Model	Word Accuracy (using 200 tied state)	Word Accuracy (using 500 tied state)	Word Accuracy (using 1000 tied state)
ts03	Bigram	32.9%	37%	29.7%
ts04	Bigram	23.5%	41.6%	28.5%
Average Result		28.2%	39.3%	29.1%

Table 5. 4 Speaker Independent Using 8 Gaussian Mixture with Skip

As we can see from table 5.4 the recognition result obtained for speaker independent test data set using the same parameter of speaker dependent test data set experiment done before we obtains 28.2%, 39.3% and 29.1% for 200, 500, and 1000 tied state values respectively. Speaker independent test data set performed better at 500 tied state for bigram language model.

The result of experiment for trigram language model find in table 5.5.

Speaker	Language Model	Word Accuracy (using 200 tied state)	Word Accuracy (using 500 tied state)	Word Accuracy (using 1000 tied state)
ts03	Tri-gram	30.6%	35.6%	28.8%
ts04	Tri-gram	19.5%	21.1%	21.1%
Average Result		25.8%	30.0%	25.1%

Table 5. 5 Speaker Independent Using 8 Gaussian Mixture with Skip

As we can see from table 5.5 results word accuracy 25.8%, 30.0% and 25.1 obtained using speaker independent test data set for trigram language at 200, 500 and 1000 tied state value. From the above two tables 5.4 and 5.5 results, for speaker independent test data set the maximum accuracy

obtained at 500 tied state. When we compare the results of the two tables regarding language models used the highest word accuracy obtained for bigram language model.

In this phase experimentation the highest word accuracy recognized in table 5.2 for speaker dependent test data that was 91.7% word accuracy. For speaker independent test data set the recognizer performs better as result obtained in table 5.4 that was 39.3% word accuracy.

However, even if we obtain better recognition result at different value of tied state for speaker dependent and independent test data set, for both test data sets the recognizer performs better for bigram language model. In above with skip transition topology phases, recognizer performs better using bigram language for both test data sets, we assume that this is because of our language model text data is small and small training data set.

Phase II Experimentation

In this phase the first experiment conducted for speaker dependent, using different parameter of acoustic model for bigram language model. The acoustic model is developed without skip transition topology and 8 Gaussian mixtures is used for this experimentation.

Table 5.6 shows the result obtained from this experiment.

Speaker Code	Language Model	Word Accuracy (using 200 tied state)	Word Accuracy (using 500 tied state)	Word Accuracy (using 1000 tied state)
ts01	Bigram	65.2%	86.1%	94.8%
ts02	Bigram	59.2%	81.1%	91.2%
Average Result		62.2%	83.6%	93%

Table 5. 6 Speaker Dependent Using 8 Gaussian Mixture without Skip

As we can see from table 5.6 results, we obtains 62.2%,83.6% and 93% word accuracy for 200,500 and 1000 tied state. The recognizer performs better at 1000 tied state.

Table 5.7 shows the recognition result obtained in case of trigram language model.

Speaker Code	Language Model	Word Accuracy (using 200 tied state)	Word Accuracy (using 500 tied state)	Word Accuracy (using 1000 tied state)
ts01	Tri-gram	62.6%	80.9%	91.3%
ts02	Tri-gram	55.8%	82.5%	89.1%
Average Result		59.2%	81.7%	90.2%

Table 5. 7 Speaker Dependent Using 8 Gaussian Mixture without Skip

As we can see from table 5.6 and 5.7 results, in this study speaker dependent test data set performs better at 1000 tied state for both bigram and trigram language model. And also bigram language model performs better again at without skip transition topology.

The second experiment is done for speaker independent using the same parameter of speaker dependent experiment mentioned above at without skip transition topology.

Speaker	Language Model	Word Accuracy (using 200 tied state)	Word Accuracy (using 500 tied state)	Word Accuracy (using 1000 tied state)
ts03	Bigram	36.4%	40.6%	32.7%
ts04	Bigram	29.2%	46.6%	29.5%
Average Result		32.8%	43.6%	31.1%

Table 5. 8 Speaker Independent Using 8 Gaussian Mixture without Skip

As we can observe from table 5.8 result, the experimentation for speaker independent test data set using the same parameter of speaker dependent experimentation is very less when we compare with speaker dependent test data set. This is because of the test data sets for speaker dependent are found in training data set.

And also in case of speaker independent experiment we got better performance at 500 tied state.

Table 5.9 shows the recognition result obtained for speaker independent using trigram LM and tied state acoustic model parameters.

Speaker	Language Model	Word Accuracy (using 200 tied state)	Word Accuracy (using 500 tied state)	Word Accuracy (using 1000 tied state)
ts03	Tri-gram	31.7%	38.8%	30.8%
ts04	Tri-gram	27.9%	32.6%	27.6%
Average Result		29.8%	35.7%	29.2%

Table 5. 9 Speaker Independent Using 8 Gaussian Mixture without Skip

As we can observe from table 5.9 result for speaker independent test data set using trigram LM we obtains 29.8%, 35.7% and 29.2% word accuracy. The recognizer performs better at 500 tied state.

From this two experimentation phases, this study large vocabulary continuous speech recognition system for Afaan Oromo using HMM performs the highest word accuracy for speaker dependent test set at 1000 tied state by using bigram language model.

We used Context Dependent acoustic model and 8-Gaussian mixture for all experimentation. In this study the maximum word accuracy obtained were 93% and 43.6% respectively, for speaker dependent and speaker independent. When we compare results obtained in this study with pervious result obtained for continuous read speech Afaan Oromo automatic speech recognition we obtained better recognition result 93% comparing to [15] pervious works in which 89.459% word accuracy obtained for speaker dependent test data set.

From the above experimentation results bigram LM performed better in case of both speaker dependent test sets and speaker independent test sets as compare to trigram LM. Bigram LM performed the highest word accuracy 93% for speaker dependent test set at without skip transition topology and 43.6% word accuracy for speaker independent test set at without skip transition

topology. This was due to the size of data, if some states skipped there are phones which were reduced and this causes decrement in data size.

Regarding, tied state values the best recognition result obtained at 1000 and 500 tied state values for speaker dependent test data set and speaker independent test data set. However, even if the recognition performance of trigram is less than bigram in every experimentation phase's trigram language model also performs better for speaker dependent test data set and speaker independent test data set at 1000 and 500 tied state respectively.

To the best of our knowledge there is no any study done before on the development of speaker independent recognizer for Afaan Oromo continuous read speech. As first study the result obtained for speaker independent recognizer is promising. We obtains 43.6% word accuracy for speaker independent recognizer. This result obtained from one hour training data we assume that this result will increase as training data set and language model data set increases. Speaker independent LVCSR requires large number of training data compared to speaker dependent ASR systems.

In general, large vocabulary continuous speech recognition system for Afaan Oromo using HMM model performed better than the previous done by Kassahun [15] in terms of vocabulary size, training data set size and word accuracy recognized for speaker dependent test data set. The large vocabulary continuous speech recognition system for Afaan Oromo language were not done before. As a first study the result obtained for speaker independent test data set were also promising.

5.7 Challenges

There are a lot of challenges we face regarding resources in different stages across the study. The data preparation stage was the most challenging task for our study. As this is the pioneer study in the area of large vocabulary continuous speech recognition system for Afaan Oromo language, there was no prepared read speech corpus for this purpose that we can use for training and testing, so we developed it from the scratch.

The other challenge is labeling and segmenting read speech data because of unclear boundary of read speech data uttered by speakers and co-articulation effects.

And also the text data we used for language model and transcription was collected from different Afaan Oromo sources manually. Text normalization is one the most challenge we faced during this study.

Finally we faced another challenges during pronunciation dictionary development because of lack of literatures related to speech recognition especially on Afaan Oromo. It is impossible to get articles/books on Afaan Oromo speech recognition. Besides, there is lack of books that deal with Afaan Oromo grammatical and semantic structures in relation with speech recognition. Moreover, it is difficult to get reference books on speech recognition and the techniques and tools used. Most of the books on speech recognition are too expensive to buy.

CHAPTER 6

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

With this study, the possibility of developing Large Vocabulary Continuous Speech Recognition System for Afaan Oromo language has been explored.

We conduct a series of experiment to identify the factors that contribute to increase the performance for developing Large Vocabulary Continuous Speech Recognition System for Afaan Oromo. The experimentation is conducted in two major phases with different acoustic model parameters. In this study, we considered Gaussian mixture weight, tied state value, transition topology for acoustic model and also bigram and trigram language model. To improve the recognizer performance we train acoustic model periodically by tune the acoustic model parameters. In addition to acoustic model the language model also have an effect on recognizer performance. To improve the recognizer performance we used 10000 text corpus for language model and also we have done additive or Laplace smoothing techniques.

We have tested the recognizer by using two types of test data set. The first is speaker dependent test set. The second is speaker independent test set. For both cases we used 100 utterances which uttered by two speakers. Each speaker uttered 50 utterance or sentences.

For this study, among the approaches of ASR systems the Statistical (stochastic) approach was used and the HMM has been implemented for modeling, CMU sphinx tool kit and different tools which are compatible with the approach and modeling technique we have preferred were implemented accordingly.

The text corpus data we used in this study were created by collecting from different Afaan Oromo sources (Fana Broadcasting Corporation (FBC), Voice of America (VOA), Books, Kassahun [15] and Teferi [23] transcription texts which contributed to inclusion all Afaan Oromo phones) used for both transcription file and language model. In addition to the corpus data two type of language model is used bi-gram and tri-gram language model. To train the acoustic model we used mono-

phone sub-word units. The training has been done using the latest version of CMU sphinx which is sphinx train pre-alpha.

Before starting acoustic model training we have prepared all data's that are used for training acoustic model. The data's that used for training acoustic model of sphinx trainer are pronunciation diction, phones, fillers, training transcription file and wav file. We put them in our database AOASR sub-folder etc and wav. The training has been done first by copying the path of our database to the training configuration file by using `sphinxtrain -t AOASR` setup command on Linux terminal since we used Linux operating system 16.04 LTS for this study.

In order to improve our recognizer accuracy we have refined our models, as a result we have developed context dependent tri-phones, we have increased the number of Gaussian mixture, we have used 200, 500 and 1000 tied state (senones) values, with skip and without skip transition topology. By starting with single Gaussian mixture, we have increased the number of Gaussian mixture in the process of refining our models and we have got better performance at 8th Gaussian mixture

The experimentation is held using the continuous read speech of Afaan Oromo corpus. The experimentation were done in two distinct phases. The first phase experimentation were done using with skip transition topology, context dependent tri-phone and using 8 Gaussian mixture. From the experimentation result bigram LM performed the highest word accuracy 93% for speaker dependent test set at without skip transition topology and 43.6% word accuracy for speaker independent test set at without skip transition topology.

From the experimentation, speaker dependent performed better at 1000 tied state values while speaker independent performed better at 500 tied state.

From the experimentation result we can see that increasing the training data and language model do not a guarantee to increase the performance of recognizer. In addition to corpus size we have to train acoustic model by using different acoustic model parameters. In this study we trained our acoustic model by Gaussian mixture, tied state, acoustic model transition topology and acoustic model types.

6.2 Recommendation

From the experimental results and what has been learned throughout the course of this research, we would like to forward our recommendations and future works with the following paragraphs.

The dictionary used for this study is canonical pronunciation dictionary, but there are the same words that were uttered differently by different speakers, dialect variation. Therefore it is better to use alternative pronunciation dictionary in order to more improve recognizer performance.

There is the possibility to get better recognition result by using data which is free from noise, which transcribed well so the features is extracted well it have a significant role in performance, so large corpus and clear data might be increase the performance.

The researcher has to give attention while preparing the data, if the data which is prepared by the researcher do have so much noise it highly degrade the performance of the recognizer this issues should be critically concerned.

Large corpus should be considered in the future works. As a result of the increase in corpus the need for other requirements in the capability of the machine to run those data is also important.

In addition to acoustic model, language model also have high effect on recognition performance. In this study we used additive or Laplace smoothing, we to use other language smoothing techniques and comparing which perform best result.

Lastly as this research is conducted on read speech, the need to consider spontaneous speech, speeches from TV broadcast, Radio and conference halls might be considered in the future works.

REFERENCES

- [1] Lawrence Rabiner, Biing-Hwang, Fundamentals of speech recognition, New jersey: Printice Hall, 1993.
- [2] T. c. Luh, "Speaker Independent Speech Recognition Using Neural Network," MSc thesis University Putra, Malaysia , 2004.
- [3] M.A.Anusuya and S.K.Katti, "Speech Recognition by Machine: A Review," (*IJCSIS*) *International Journal of Computer Science and Information Security*, vol. vol6, 2009.
- [4] R. K. Aggarwal, and M. Dave, "Acoustic Modeling Problem for Automatic Speech Recognition System," *International journal Speech Technology, Springer*, vol. 14, no. 2, 2011.
- [5] S.E.Levinson, "Structural Methods in Automatic Speech Recognition," in *Structural Methods in Automatic Speech Recognition*, 1985, pp. 1625-1694.
- [6] Muhirwe Jackson, "Automatic Speech Recognition: Human Computer Interface for Kinyarwanda Language," M.Sc. Thesis ,Makerere University, 2005.
- [7] Mikko Kurimo, Kalle Palomäki, Janne Pytkönen, Ville T. Turunen, Sami Virpioja, Ulpu Remes, Heikki Kallasjoki, Reima Karhila, TeemuRuokolainen, Tanel Alumäe, Sami Keronen, André Mansikkaniemi, Peter Smit, Rama Sanand Doddipatla, Seppo Enarvi, "Automatic Speech Recognition," p. 114.
- [8] Bantegize Addis, "BRANA (ብራና): APPLICATION OF AMHARIC SPEECH RECOGNITION SYSTEM for Dictation in judicial domain," Master's thesis, Gondar University, Gondar, 2015.
- [9] Vrinda ,and Chander.S, "SPEECH RECOGNITION SYSTEM FOR ENGLISH LANGUAGE," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. Vol. 2, no. Issue 1, January 2013.
- [10] Sanjib.D, "Speech Recognition Technique: A Review," *International Journal of Engineering Research and Applications*, vol. Vol. 2, no. Issue 3, May-Jun 2012.

- [11] Solomon.T and Wolfgang, "Syllable-Based Speech Recognition for Amharic," in *Proceedings of the 5th Workshop on Important Unresolved Matters,prague, Czech Republic*, 2007.
- [12] a. e. Santosh K, "A Review on Speech Recognition Technique," *International Journal of Computer Applications*, vol. Vol10, November 2010.
- [13] Woosung.K, LANGUAGE MODEL ADAPTATION FOR AUTOMATIC SPEECH RECOGNITION AND STATISTICAL MACHINE TRANSLATION, October 2004.
- [14] B.H. Juang & Lawrence R. Rabiner, "Automatic Speech Recognition – A Brief History of the Technology," 2004.
- [15] Kassahun Gelana, "A continuous, speaker independent Afaan Oromo speech recognizer," Master's Thesis, Addis Ababa University, Addis Ababa, 2010.
- [16] Solomon Berhanu, "Isolated Amharic Consonant-Vowel (CV) Syllable Recognition: An Experiment Using the Hidden Markov Model," M.Sc. Thesis, Addis Ababa University, Addis Ababa , 2001.
- [17] Kinfe Tadesse, "Sub-Word Based Amharic Word Recognition: An Experiment Using Hidden Markov Model (HMM)," M. Sc. Thesis, Addis Ababa University, Addis Ababa, 2002.
- [18] Zegaye Seifu, "HMM based large vocabulary, speaker in-dependent, continuousAmharic speech recognizer," M.Sc Thesis, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, 2003.
- [19] Martha Yifiru Tachbelie, "Application of Amharic Speech Recognition system to command and control computer: an experiment with Microsoft word," Master's Thesis, Addis Ababa University, Addis Ababa, 2003.
- [20] Hussien Seid, "A Speaker Independent Continuous Amharic Speech Recognizer: an Experiment using hybrid (HMM-ANN)," M. Sc. Thesis, Addis Ababa University, Addis Ababa, 2004.
- [21] Asafa Jalata, "The present and future of the Oromoo people," Journal of the Oromoo literature, a scholarly publishing initiatives, Minnesota, 2010.
- [22] Ashenafi Demissie, "A Speech Recognition System for Afaan Oromo," MSc thesis, Addis Ababa University, Addis Ababa, 2009.

- [23] Teferi Kebebew, "Speech Recognition for Afan Oromo Using Hybrid," Master's Thesis, Addis Ababa University, Addis Ababa, 2010.
- [24] Dr.V.Ajantha Devi , Ms.V.Suganya, "An Analysis on Types of Speech Recognition and Algorithms," *International Journal of Computer Science Trends and Technology (IJCST)*, vol. IV, no. 2, pp. 350-355, 2016.
- [25] Yongmei Shi, "An Investigation of linguistic information for speech recognition error detection," Phd Dissertation, University of Maryland, Baltimore, 2008.
- [26] J. A. Markowitz, *Using Speech Recognition*, Upper Saddle River, New Jersey: Prentice Hall, 1996.
- [27] Deller, J.R. Jr., Hansen, J.H.L. and Proakis, J.G. , *Discrete-time Processing of Speech*, New York, USA: Macmillan Publishing Company, 2000.
- [28] Solomon Teferra Abate, Martha Yifiru Tachbelie, Wolfgang Menzel, "Amharic Speech Recognition: Past, Present and Future," in *Proceedings of the 16th International Conference of Ethiopian Studies*, Trondheim , 2009.
- [29] Namrata.D, "Feature Extraction Methods LPC, PLP and MFCC," *INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY*, vol. Volume 1, no. Issue VI, July 2013.
- [30] Tuka.Al, *Lexical and Language Modeling of Diacritics and Morphemes in Arabic Automatic Speech Recognition*, February 2014.
- [31] Jurafsky, and Martin , *Speech and language processing An introduction to Natural language processing, computational linguistics , and speech recognition*, Second ed., New jersey: Printice Hall, 2007.
- [32] Wiqas Ghai and Navdeep Singh, "Literature Review on Automatic Speech Recognition," *International Journal of Computer Applications*, vol. Vol41, March 2012.
- [33] a. e. R. Thangarajan, "Word and Triphone Based Approaches in Continuous Speech," vol. Volume 4, no. Issue 3, March 2008.
- [34] Tilahun Gamta, "Online Oromo Dictionary," 29 8 1992. [Online]. Available: <http://oromodictionary.com/articles/article.php?id=7>. [Accessed 5 January 2016].

- [35] Ethnomed.org, "ethnoMED," 1 December 2002. [Online]. Available: <https://ethnomed.org/culture/oromo/oromo-alphabets-and-sounds-sagaleewani-fi-loqoda>. [Accessed 13 August 2015].
- [36] Girma Debele, "Afan Oromo news text summarizer," Masters' Thesis, Addis Ababa University, Addis Ababa, 2010.
- [37] I. Thompson, "About World Languages," 23 March 2015. [Online]. Available: <http://aboutworldlanguages.com/oromo>. [Accessed 5 January 2016].
- [38] Mylanguage.org, "My Language," 2011. [Online]. Available: http://www.mylanguages.org/learn_oromo.php. [Accessed 13 August 2015].
- [39] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, Joe Woelfel, "Sphinx-4: A Flexible Open Source Framework for Speech Recognition," SUN MICROSYSTEMS INC, 2004.

Appendix A: Some Afaan Oromo Text Corpus

<s> aadaan calaqkee jiruu fi jireenya saba tokkooti </s> (tr01-001)
<s> saba dinagdeen guddatee fi siyaasaan qaroomme </s> (tr01-002)
<s> bakka hedduutti addaan qoodnee ilaalu ni dandeenya </s> (tr01-003)
<s> waaqa nu uumetti bulla </s> (tr01-004)
<s> ayyaani hiikoo waan hedduu of keessaa qaba </s> (tr01-005)
<s> waaqeffannaa keessatti waanti hundi ayyaana mataa isaa qaba </s> (tr01-006)
<s> gaaddidduu dhabeessa taha </s> (tr01-007)
<s> safuun waan baayyee of keessaa qaba </s> (tr01-008)
<s> cubbuu hojjachuun immoo nagaa namootaa yookiin hawaasaa booressuu dha </s> (tr01-009)
<s> ateeteen ayyaana haadhaati </s> (tr01-010)
<s> yeroo fi akkaataan itti kabajamu naannoo naannootti xiqqoo addummaa qabatus jifuu fi duudhaan guutamuu qabu tokkuma </s> (tr01-011)
<s> ayyaanni irreechaa waggaatti yeroo lama birraa fi arfaasaa kabajama </s> (tr01-012)
<s> gumii waaqeffattoota addunyaa </s> (tr01-013)
<s> maallima warri jedhaman gosa oromoo keessaa tokko </s> (tr01-014)
<s> kennaa addaa warra qabanii dha </s> (tr01-015)
<s> marabbaan kan waaqaati </s> (tr01-016)
<s> maqaa dhahuun nidandahama </s> (tr01-017)
<s> gotoota oromoo haa yaadannuu </s> (tr01-018)
<s> waggaa saddet guutuu ishee eegaa ture </s> (tr01-019)
<s> qabsaawaan ni kufaa qabsoon itti fufa </s> (tr01-020)
<s> waldaa artistoota oromoo </s> (tr01-021)
<s> ergamtoota diinaa afaan qawweetiin nu ajjeesaa jiraniif hin bitamnu </s> (tr01-022)
<s> barattoota mana barumsaa olaanaa sadarkaa lammaffaa amboo </s> (tr01-023)
<s> sadarkaan gadaa tokko waggaa saddeet qaba </s> (tr01-024)
<s> oromoon damee kuush jalatti argama </s> (tr01-025)
<s> sirni gadaa walqixxummaa saalaa mirkaneessuu irrattis mudaa akka hin qabne seenaan ragaa dha </s> (tr01-026)
<s> balaa kanaan namoonni sadii miidhamanii mana yaalaa akiraa keessatti argamutti akka yaalaman taasifamuu niyaadatama </s> (tr01-027)
<s> jiha caamsaa keessa akka ifa taasifamu ibsaniiru </s> (tr01-028)
<s> gibtsi lammiiwwanshee garee dhaadheessaa islaamaatiin ajjeefamaniif gadda torbaan tokkoo labsite </s> (tr01-029)
<s> mareen seenaa obbo mallas zeenaawii irratti xiyyeeffate taasifame </s> (tr01-030)
<s> hojiin suphaa pilaazmaa manneen barnootaa oromiyaa duubatti harkifate </s> (tr01-031)
<s> ejensichi dhaabbilee sagal seeraan ala hojjetan jedheen hojii akka dhaabsise beeksise </s> (tr01-032)
<s> ummanni itiyooophiyaa marti filannoo marsaa shanaffaa milkeessuuf shoorasaa akka gumaachuuf gaafatame </s> (tr01-033)
<s> muummichi ministeeraa obbo hayilamaariyaam dassaalany gabaasa dhiyeessan </s> (tr01-034)
<s> itiyooophiyaan wayita ammaa adeemsa guddina saffisaa irratti akka argamtu ibsame </s> (tr01-035)
<s> qusannaa biyya keessa dabaluuf carraaqqiin taasifamaa jira </s> (tr01-036)

<s> israayel lammileen ishee yuuroop keessa jiraatu </s> (tr01-037)
 <s> itiyoophiyaafi jibuutiin koree waliin wajjinii hundeessuuf waliigalan </s> (tr01-038)
 <s> tajaajila oduu itiyoophiyaatu gabaase </s> (tr01-039)
 <s> koreen waliin wajjinii biyyaalee lamaanii ni eegama </s> (tr01-040)
 <s> piroojaktiin ijaarsa daandii baaburaa maqalee waldiyaa haraha jalqabamu bor ifatti labsama </s> (tr01-041)
 <s> magaalaa burraayyuutti balaan tiraafikaa suukaneessaan mudate </s> (tr01-042)
 <s> lubbuun namoota konkolaataa keessa turanii akka darbe </s> (tr01-043)
 <s> inni kaleessa finfinnee deeme </s> (tr01-044)
 <s> oolmaan inni naaf oole guddaa dha </s> (tr01-045)
 <s> garaakootii hin badu </s> (tr01-046)
 <s> qabsoon keenya guyyumaa guyyaan fuladuratti tarkaanfataa jira </s> (tr01-047)
 <s> balaawwan akka kanaa saffisaan konkolaachisuun mudata jedhan inspeektar tsaggaayee baayyuu </s> (tr01-048)
 <s> dhaabbanni bosonaafi bineensota bosona oromiyaa bosona lafa hektaara miliyoona lama caalurra jiru daangesse </s> (tr01-049)
 <s> obbo hayilmaariyaam mana maree federeeshinii galateeffataniiru </s> (tr01-050)

Appendix B: Afaan Oromo Canonical Pronunciation Dictionary

aadaa	aa d aa
aadaadhuma	aa d aa dh u m a
aadaafi	aa d aa f i
aadaan	aa d aa n
aafrikaa	aa f r i k aa
aananii	aa n a n ii
aangoo	aa n g oo
aangoon	aa n g oo n
aannan	aa n a n
aannani	aa n a n i
aannanii	aa n a n ii
aaree	aa r ee
aartii	aa r t ii
aartisti	aa r t i s t i
abaaboon	a b aa b oo n
abbaa	a bb aa
abbaan	a bb aa n
abbaasaaf	a bb aa s aa f
abbootii	a bb oo t ii
abdii	a b d ii
abdiin	a b d ii n
abidda	a b i dd a

abiddaa	a b i dd aa
abjuu	a b j uu
achi	a ch i
adaamaa	a d aa m aa
adda	a dd a
addaa	a dd aa
addaan	a dd aa n
addaatiin	a dd aa t ii n

Appendix C: Afaan Oromo Phone Set

SIL	dh	j	ns	s	ww
a	e	jj	ny	sh	x
aa	ee	k	o	ss	xx
b	f	kk	oo	t	y
bb	ff	l	p	ts	yy
c	g	ll	ph	tt	z
cc	gg	m	q	u	
ch	h	mm	qq	uu	
d	i	n	r	v	
dd	ii	nn	rr	w	

Appendix D: Training Configuration

Configuration script for sphinx trainer -*-mode:Perl-*-

\$CFG_VERBOSE = 1; # Determines how much goes to the screen.

These are filled in at configuration time

\$CFG_DB_NAME = "AOASR";

Experiment name, will be used to name model files and log files

\$CFG_EXPTNAME = "\$CFG_DB_NAME";

Directory containing SphinxTrain binaries

\$CFG_BASE_DIR = "/home/dura/AO/AOASR";

\$CFG_SPHINXTRAIN_DIR = "/usr/local/lib/sphinxtrain";

\$CFG_BIN_DIR = "/usr/local/libexec/sphinxtrain";

\$CFG_SCRIPT_DIR = "/usr/local/lib/sphinxtrain/scripts";

Audio waveform and feature file information

\$CFG_WAVFILES_DIR = "\$CFG_BASE_DIR/wav";

\$CFG_WAVFILE_EXTENSION = 'wav';

```

$CFG_WAVFILE_TYPE = 'mswav'; # one of nist, mswav, raw
$CFG_FEATFILES_DIR = "$CFG_BASE_DIR/feat";
$CFG_FEATFILE_EXTENSION = 'mfc';

# Feature extraction parameters
$CFG_WAVFILE_SRATE = 16000.0;
$CFG_NUM_FILT = 25; # For wideband speech it's 25, for telephone 8khz reasonable value is
15
$CFG_LO_FILT = 130; # For telephone 8kHz speech value is 200
$CFG_HI_FILT = 6800; # For telephone 8kHz speech value is 3500
$CFG_TRANSFORM = "dct"; # Previously legacy transform is used, but dct is more accurate
$CFG_LIFTER = "22"; # Cepstrum lifter is smoothing to improve recognition
$CFG_VECTOR_LENGTH = 13; # 13 is usually enough

$CFG_MIN_ITERATIONS = 1; # BW Iterate at least this many times
$CFG_MAX_ITERATIONS = 10; # BW Don't iterate more than this, somethings likely wrong.

# (none/max) Type of AGC to apply to input files
$CFG_AGC = 'none';
# (current/none) Type of cepstral mean subtraction/normalization
# to apply to input files
$CFG_CMN = 'current';
# (yes/no) Normalize variance of input files to 1.0
$CFG_VARNORM = 'no';
# (yes/no) Train full covariance matrices
$CFG_FULLVAR = 'no';
# (yes/no) Use diagonals only of full covariance matrices for
# Forward-Backward evaluation (recommended if CFG_FULLVAR is yes)
$CFG_DIAGFULL = 'no';
# (yes/no) Perform vocal tract length normalization in training. This
# will result in a "normalized" model which requires VTLN to be done
# during decoding as well.
$CFG_VTLN = 'no';
# Starting warp factor for VTLN
$CFG_VTLN_START = 0.80;
# Ending warp factor for VTLN
$CFG_VTLN_END = 1.40;
# Step size of warping factors
$CFG_VTLN_STEP = 0.05;
# Directory to write queue manager logs to

```

```

$CFG_QMGR_DIR = "$CFG_BASE_DIR/qmanager";
# Directory to write training logs to
$CFG_LOG_DIR = "$CFG_BASE_DIR/logdir";
# Directory for re-estimation counts
$CFG_BWACCUM_DIR = "$CFG_BASE_DIR/bwaccumdir";
# Directory to write model parameter files to
$CFG_MODEL_DIR = "$CFG_BASE_DIR/model_parameters";
# Directory containing transcripts and control files for
# speaker-adaptive training
$CFG_LIST_DIR = "$CFG_BASE_DIR/etc";
# Decoding variables for MMIE training
$CFG_LANGUAGEWEIGHT = "11.5";
$CFG_BEAMWIDTH      = "1e-100";
$CFG_WORDBEAM       = "1e-80";
$CFG_LANGUAGEMODEL  = "$CFG_LIST_DIR/$CFG_DB_NAME.lm.bin";
$CFG_WORDPENALTY    = "0.2";
# Lattice pruning variables
$CFG_ABEAM          = "1e-50";
$CFG_NBEAM          = "1e-10";
$CFG_PRUNED_DENLAT_DIR = "$CFG_BASE_DIR/pruned_denlat";
# MMIE training related variables
$CFG_MMIE = "no";
$CFG_MMIE_MAX_ITERATIONS = 5;
$CFG_LATTICE_DIR = "$CFG_BASE_DIR/lattice";
$CFG_MMIE_TYPE = "rand"; # Valid values are "rand", "best" or "ci"
$CFG_MMIE_CONSTE = "3.0";
$CFG_NUMLAT_DIR = "$CFG_BASE_DIR/numlat";
$CFG_DENLAT_DIR = "$CFG_BASE_DIR/denlat";
# Variables used in main training of models
$CFG_DICTIONARY = "$CFG_LIST_DIR/$CFG_DB_NAME.dic";
$CFG_RAWPHONEFILE = "$CFG_LIST_DIR/$CFG_DB_NAME.phone";
$CFG_FILLERDICT = "$CFG_LIST_DIR/$CFG_DB_NAME.filler";
$CFG_LISTOFFILES = "$CFG_LIST_DIR/${CFG_DB_NAME}_train.fileids";
$CFG_TRANSCRIPTFILE = "$CFG_LIST_DIR/${CFG_DB_NAME}_train.transcription";
$CFG_FEATPARAMS = "$CFG_LIST_DIR/feat.params";
# Variables used in characterizing models

$CFG_HMM_TYPE = '.cont.'; # Sphinx 4, PocketSphinx
#$CFG_HMM_TYPE = '.semi.'; # PocketSphinx
#$CFG_HMM_TYPE = '.ptm.'; # PocketSphinx (larger data sets)

```

```

if (($CFG_HMM_TYPE ne ".semi.")
    and ($CFG_HMM_TYPE ne ".ptm.")
    and ($CFG_HMM_TYPE ne ".cont.")) {
    die "Please choose one CFG_HMM_TYPE out of '.cont.', '.ptm.', or '.semi.', " .
        "currently $CFG_HMM_TYPE\n";
}
# This configuration is fastest and best for most acoustic models in
# PocketSphinx and Sphinx-III. See below for Sphinx-II.
$CFG_STATESPERHMM = 3;
$CFG_SKIPSTATE = 'no';
if ($CFG_HMM_TYPE eq '.semi.') {
    $CFG_DIRLABEL = 'semi';
# Four stream features for PocketSphinx
    $CFG_FEATURE = "s2_4x";
    $CFG_NUM_STREAMS = 4;
    $CFG_INITIAL_NUM_DENSITIES = 256;
    $CFG_FINAL_NUM_DENSITIES = 256;
    die "For semi continuous models, the initial and final models have the same density"
        if ($CFG_INITIAL_NUM_DENSITIES != $CFG_FINAL_NUM_DENSITIES);
} elsif ($CFG_HMM_TYPE eq '.ptm.') {
    $CFG_DIRLABEL = 'ptm';
# Four stream features for PocketSphinx
    $CFG_FEATURE = "s2_4x";
    $CFG_NUM_STREAMS = 4;
    $CFG_INITIAL_NUM_DENSITIES = 64;
    $CFG_FINAL_NUM_DENSITIES = 64;
    die "For phonetically tied models, the initial and final models have the same density"
        if ($CFG_INITIAL_NUM_DENSITIES != $CFG_FINAL_NUM_DENSITIES);
} elsif ($CFG_HMM_TYPE eq '.cont.') {
    $CFG_DIRLABEL = 'cont';
# Single stream features - Sphinx 3
    $CFG_FEATURE = "1s_c_d_dd";
    $CFG_NUM_STREAMS = 1;
    $CFG_INITIAL_NUM_DENSITIES = 1;
    $CFG_FINAL_NUM_DENSITIES = 8;
    die "The initial has to be less than the final number of densities"
        if ($CFG_INITIAL_NUM_DENSITIES > $CFG_FINAL_NUM_DENSITIES);
}
# Number of top gaussians to score a frame. A little bit less accurate computations
# make training significantly faster. Uncomment to apply this during the training

```

```

# For good accuracy make sure you are using the same setting in decoder
# In theory this can be different for various training stages. For example 4 for
# CI stage and 16 for CD stage
# $CFG_CI_TOPN = 4;
# $CFG_CD_TOPN = 16;
# (yes/no) Train multiple-gaussian context-independent models (useful
# for alignment, use 'no' otherwise) in the models created
# specifically for forced alignment
$CFG_FALIGN_CI_MGAU = 'no';
# (yes/no) Train multiple-gaussian context-independent models (useful
# for alignment, use 'no' otherwise)
$CFG_CI_MGAU = 'no';
# (yes/no) Train context-dependent models
$CFG_CD_TRAIN = 'yes';
# Number of tied states (senones) to create in decision-tree clustering
$CFG_N_TIED_STATES = 1000;
# How many parts to run Forward-Backward estimation in
$CFG_NPART = 1;

# (yes/no) Train a single decision tree for all phones (actually one
# per state) (useful for grapheme-based models, use 'no' otherwise)
$CFG_CROSS_PHONE_TREES = 'no';

# Use force-aligned transcripts (if available) as input to training
$CFG_FORCEDALIGN = 'no';

# Use a specific set of models for force alignment. If not defined,
# context-independent models for the current experiment will be used.
$CFG_FORCE_ALIGN_MODELDIR =
"$CFG_MODEL_DIR/$CFG_EXPTNAME.falign_ci_$CFG_DIRLABEL";

# Use a specific dictionary and filler dictionary for force alignment.
# If these are not defined, a dictionary and filler dictionary will be
# created from $CFG_DICTIONARY and $CFG_FILLERDICT, with noise words
# removed from the filler dictionary and added to the dictionary (this
# is because the force alignment is not very good at inserting them)

# $CFG_FORCE_ALIGN_DICTIONARY =
"$ST::CFG_BASE_DIR/falignout$ST::CFG_EXPTNAME.falign.dict";

```

```

# $CFG_FORCE_ALIGN_FILLERDICT =
"$ST::CFG_BASE_DIR/falignout/$ST::CFG_EXPTNAME.falign.fdict";

# Use a particular beam width for force alignment. The wider
# (i.e. smaller numerically) the beam, the fewer sentences will be
# rejected for bad alignment.
$CFG_FORCE_ALIGN_BEAM = 1e-60;

# Calculate an LDA/MLLT transform?
$CFG_LDA_MLLT = 'no';
# Dimensionality of LDA/MLLT output
$CFG_LDA_DIMENSION = 29;

# This is actually just a difference in log space (it doesn't make
# sense otherwise, because different feature parameters have very
# different likelihoods)
$CFG_CONVERGENCE_RATIO = 0.1;

# Queue::POSIX for multiple CPUs on a local machine
# Queue::PBS to use a PBS/TORQUE queue
$CFG_QUEUE_TYPE = "Queue";

# Name of queue to use for PBS/TORQUE
$CFG_QUEUE_NAME = "workq";

# (yes/no) Build questions for decision tree clustering automatically
$CFG_MAKE_QUESTS = "yes";
# If CFG_MAKE_QUESTS is yes, questions are written to this file.
# If CFG_MAKE_QUESTS is no, questions are read from this file.
$CFG_QUESTION_SET =
"${CFG_BASE_DIR}/model_architecture/${CFG_EXPTNAME}.tree_questions";
#$CFG_QUESTION_SET = "${CFG_BASE_DIR}/linguistic_questions";

$CFG_CP_OPERATION =
"${CFG_BASE_DIR}/model_architecture/${CFG_EXPTNAME}.cpmeanvar";

# Configuration for grapheme-to-phoneme model
$CFG_G2P_MODEL= 'no';

# Configuration script for sphinx decoder

```

```

# Variables starting with $DEC_CFG_ refer to decoder specific
# arguments, those starting with $CFG_ refer to trainer arguments,
# some of them also used by the decoder.

$DEC_CFG_VERBOSE = 1;          # Determines how much goes to the screen.

# These are filled in at configuration time

# Name of the decoding script to use (psdecode.pl or s3decode.pl, probably)
$DEC_CFG_SCRIPT = 'psdecode.pl';

$DEC_CFG_EXPTNAME = "$CFG_EXPTNAME";
$DEC_CFG_JOBNAME = "$CFG_EXPTNAME"."_job";

# Models to use.
$DEC_CFG_MODEL_NAME =
"$CFG_EXPTNAME.cd_${CFG_DIRLABEL}_${CFG_N_TIED_STATES}";

$DEC_CFG_FEATFILES_DIR = "$CFG_BASE_DIR/feat";
$DEC_CFG_FEATFILE_EXTENSION = '.mfc';
$DEC_CFG_AGC = $CFG_AGC;
$DEC_CFG_CMN = $CFG_CMN;
$DEC_CFG_VARNORM = $CFG_VARNORM;

$DEC_CFG_QMGR_DIR = "$CFG_BASE_DIR/qmanager";
$DEC_CFG_LOG_DIR = "$CFG_BASE_DIR/logdir";
$DEC_CFG_MODEL_DIR = "$CFG_MODEL_DIR";

$DEC_CFG_DICTIONARY = "$CFG_BASE_DIR/etc/$CFG_DB_NAME.dic";
$DEC_CFG_FILLERDICT = "$CFG_BASE_DIR/etc/$CFG_DB_NAME.filler";
$DEC_CFG_LISTOFFILES = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}_test.fileids";
$DEC_CFG_TRANSCRIPTFILE =
"$CFG_BASE_DIR/etc/${CFG_DB_NAME}_test.transcription";
$DEC_CFG_RESULT_DIR = "$CFG_BASE_DIR/result";
$DEC_CFG_PRESULT_DIR = "$CFG_BASE_DIR/presult";

# This variables, used by the decoder, have to be user defined, and
# may affect the decoder output

```

```

$DEC_CFG_LANGUAGEMODEL = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.lm.bin";
# Or can be JSGF or FSG too, used if uncommented
# $DEC_CFG_GRAMMAR = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.jsgf";
# $DEC_CFG_FSG = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.fsg";

$DEC_CFG_LANGUAGEWEIGHT = "10";
$DEC_CFG_BEAMWIDTH = "1e-80";
$DEC_CFG_WORDBEAM = "1e-40";

$DEC_CFG_ALIGN = "builtin";

$DEC_CFG_NPART = 1;          # Define how many pieces to split decode in

# This variable has to be defined, otherwise utils.pl will not load.
$CFG_DONE = 1;

return 1;

```