Addis Ababa University
College of Natural and Computational Science
School of Information Science

Amharic Connected Word Speech Recognition System for Mobile Phones

By

Bewunetu Dagne

October, 2017

ADDIS ABABA, ETHIOPIA

Addis Ababa University
College of Natural and Computational Science
School of Information Science

Amharic Connected Word Speech Recognition System for Mobile Phones

By

Bewunetu Dagne

A Thesis submitted to Addis Ababa University in partial fulfillment of the requirement for the Degree of Masters of Science in Information Science

October, 2017

ADDIS ABABA, ETHIOPIA

Amharic Connected Word Speech Recognition System for Mobile Phones

By

Bewunetu Dagne

Name and signature of members of the examining board

Name	Title	Signature	Date
_____	Chairperson	_____	_____
Martha Yifru (Ph. D)_____	Advisor	_____	_____
Solomon Tefera (Ph. D)_____	Examiner	_____	_____
Wondwossen Mulugeta (PhD)__	Examiner	_____	_____

Table of Contents

List of Tables

List of Figures

List of Acronyms

API	Application Program Interface
ASR	Automatic Speech Recognition
BSD	Berkley Software Distribution
CD	Context Dependent
CV	Consonant Vowel
DSR	Distributed Speech Recognition
ESR	Embedded Speech Recognition
GSM	Global System for Mobile Communication
HMM	Hidden Markov Model
JSGF	Java Speech Grammar Format
LTS	Long Term Support
MFCC	Mel Frequency Cepstral Coefficient
NDK	Native Development Kit
NSR	Network Speech Recognition
PC	Personal Computer
PDA	Personal Digital Assistant
PTM	Phonetically Tied Model
SDK	Software Development Kit
WDP	Word-dependent phone
WDCVS	Word Dependent Consonant Vowel Syllable
WER	Word Error Rate

Acknowledgment

First and foremost, I would like to thank my loving Creator, the almighty God, for making me a curious being who loves to explore His creation, and also for giving me the opportunity and strength to do this research work. Without Him, I would do nothing. I would also like to thank blessed virgin St. Mary for her perpetual help and prayer for me.

Second, my gratitude goes to my advisor Martha Yifiru (Ph. D) for her constructive comments, valuable suggestions and good guidance.

Last but not least, my appreciation goes to my family and all my colleagues who helped me and also gave me valuable support and advice to complete my study.

Bewunetu Dagne

October, 2017

Abstract

This study investigated ways of integrating an Amharic Automatic speech recognition (ASR) system on mobile phones and demonstrated the possibility of using a connected word Amharic ASR system that can be used to command and control mobile phone devices – devices which currently have become not only a valuable means of communication, but also a common computing device that we carry everywhere.

To this end, CMUSphinx's Pocketsphinx is used to build a connected word, offline, small vocabulary and speaker dependent Amharic ASR system. For the fact that CMUSphinx system does not support whole word models rather word-dependent phone models, in this research work, two distinct acoustic models namely, word-dependent phone (WDP) and word-dependent CV syllable (WDCVS) models are built. And a total of 36 words are used in both models to recognize the Amharic digits from 0 to 100 and a limited set of command phrases, spoken in a connected manner. A prototype Android application is also developed and used to integrate the developed acoustic models with an Android phone. To model the sequence of words acceptable in the prototype application, three different context free grammar files are created and used.

The approach used in implementing the recognizer is embedded speech recognition, or simply ESR, using the Pocketsphinx tool – an open source speech recognition toolkit optimized to run on embedded devices. The methodology employed to develop the recognizer is the most popular statistical model which is the Hidden Markov Model.

In this study, two categories of performance evaluations are carried out. The first, Category I, is conducted to determine the effect the three model types i.e. continuous, semi-continuous and PTM, have on the two models i.e. WDP and WDCVS, and the second, Category II, is conducted to determine the best of the two models on a mobile phone using the prototype application. Hence, in the second category of evaluations, the best recognition accuracies found for the two models are, 98.07% for WDCVS model and 97.58% % for the WDP model.

Thus, according to the results gained, it is highly promising to fully deploy Amharic ASR systems to command, control and perform other activities on mobile phones using Amharic language.

Key words: Amharic Speech recognition on mobile phones, Word-dependent phone model, Word-dependent CV syllable model.

Chapter One

Introduction

1.1 Background

Automatic speech recognition is a technology that allows a computer to identify the words that a person speaks into a microphone or telephone and convert it to written text. It takes an audio stream as input and turns it into a text transcription. To convert human voice into an on-screen text or machine understandable language, computer must go through several steps. Vibrations in the air are made when someone speaks and a sound is created. The system captures and filters the captured sound which let the sound to be digitized to remove unwanted noise, and to separate it into different bands of frequency. Frequency is the wavelength of the sound waves, heard by humans as differences in pitch. People don't speak at the same speed every day, because of that the sound must be adjusted to match the speed of the sample sound template that is already stored in the system's memory [1] [2].

Speech recognition allows users to provide input to an application with voice. Just like clicking with mouse, typing on a keyboard, or pressing a key on the phone keypad provides input to an application, speech recognition allows to provide input by talking. In the desktop world, the user needs a microphone to be able to do this.

The recognition process is performed by a software component known as the speech recognition engine. The primary function of a speech recognition engine is to process spoken input and translate it into text that an application understands. The application can interpret the result of recognition as a command. In this case the application is a command and control application. An example of a command and control application is one in which a caller says "check balance", and the application returns the current balance of the caller's account. If an application handles the recognized text simply as text, then it is considered a dictation application. In a dictation application, if you said "check balance," the application would not interpret the result, but simply return the text "check balance" [1].

It is a fact that computing has revolutionized the way the world works and it is also clear that, as technology is advancing, the scale of computer use is increasing and allowing the use of computers in different ways and forms. Today, we use our mobile phones to carry out most of the tasks that we used to do using our personal computers in the past. Tan and Lindberg [3] stated that, "Computing is penetrating every corner of our life: Mobile devices bring computers all over the place and networks connect everywhere to computing resources." Another report by ITU and G3ict [4] state that, Mobile communications have become ubiquitous, reaching out to the most isolated and least served communities in developed and developing countries alike.

After several decades of intensive research and with the help of increasing processing power of mobile phones just like that of the computers, the state-of-the-art speech recognition technology has already reached a level where it enables the user to control mobile phones using speech commands, to conduct dialogues and also to accurately dictate text (even when using large vocabulary continuous speech), mostly in controlled situations [5]. ASR systems on mobile phones can significantly improve the use of these devices not only by disabled, elder or illiterate people but also by everybody else, since ASR systems utilize the most simple, common and yet convenient method of interaction which is the speech, and also since every mobile phone has a microphone and speech output which can readily be used in the speech recognition and synthesis activity.

Speech Recognition System can be implemented in many different areas and methods. When it comes to integrating an ASR Systems on Mobile Devices, there are three common approaches: network speech recognition (NSR), distributed speech recognition (DSR) and embedded speech recognition (ESR). In NSR, speech signal is transmitted to the server where feature extraction, recognition and decoding are conducted. DSR adopts the client-server architecture by placing feature extraction in the client and recognition and decoding in the server. In *ESR*, all ASR processing are conducted in the target device [3]. In general, all the three methods have their own advantages and disadvantages and the choice should consider the purpose and also the environment that the speech recognition system is going to be applied or used.

Speech recognition systems can be separated in several different classes based on different parameters, *i.e.* the type of speech utterance or the vocabulary size that they have the ability to recognize. Accordingly, based on the type of speech utterance, ASR systems can recognize isolated words, connected words or words spoken continuously or spontaneously, each with different techniques and for different purposes. Rabiner and Juang [6], stated the primary goal of connected word ASR systems as, "to create a robust system capable of recognizing a fluently spoken string of words (e.g., digits) based on matching a concatenated pattern of individual words".

Research in automatic speech recognition for Amharic started in 2001 when Solomon [7] developed isolated Consonant-Vowel syllable recognition system. Since then, many researches have been done by different researchers and with different perspectives. However, when it comes to incorporating an Amharic speech recognition system to a mobile phone, an intensive investigation of literature reveals that there are only two research works done by Tinbit [8] and Mikhael [9].

The main focus of this research work is thus, to investigate an improved way of creating and integrating a connected word Amharic ASR system that can be used to command and control mobile phones using the Amharic language.

1.2 Statement of the Problem and Justification

Across the world, mobile devices, including mobile phones, laptops and PDAs, are quickly becoming our primary means of communication, information processing and also sources of entertainment. This transition has been driven by both the attraction of end users to these devices and also by the continued advancement in the processor, memory, display and wireless data network capabilities of these devices [4]. Modern mobile phones, even low-cost versions, now come with more processor and memory than desktop PCs from a few years ago. And because of these advancements, the mobile devices have become ubiquitous and people are relying on mobile phones more and more in their daily lives [10].

However, despite the ubiquity of these devices, these devices lack the necessary features to be fully used by the whole society creating a user interaction issues that can't be fully solved by extending the graphical user interface and keyboard on these small devices. Senior citizens and

people with physical disabilities often are unable to operate and utilize mobile phones as these equipments lack the necessary accessibility features or because the price of the adapted phones and services remain unaffordable [4]. And from the foreseen alternatives, speech recognition technology is, one of the promising solutions to this problem.

Though, research works to integrate speech recognition systems on mobile phones started a decade ago for most of the resourced languages like English and most European languages and have greatly enhanced the usability of the mobile phones, locally, there are only two research works done by Tinbit [8] and Mikhael [9] to integrate an ASR system with mobiles for commanding mobile phones and for phone dialing respectively, using the Amharic language.

The main focus of Tinbit's [8] work is to command mobile phones using Amharic voice, and for this, she used a small vocabulary consisting of 19 Amharic words, from which, 10 of them are used to recognize the Amharic digit words from 0 to 9 and the remaining to form command words in a connected manner i.e. "Google kefete" and "Tiri astelalf". Tinbit has adapted the English Acoustic model for Amharic language, that is, she used a cross-language adaptation technique to map or represent the Amharic language sounds/phonemes using the English phone set. And an accuracy of 72% is reported on Timbit's [8] paper after testing the application by five people.

The other research work that relates to incorporating an Amharic ASR system with a mobile phone is Mikhael [9]. Mikhael has designed a small vocabulary, speaker dependent and continuous Amharic ASR that can recognize continuously spoken digit strings and a limited set of names spoken to a PC. The researcher has developed a voice dialing application on an Android mobile phone to receive the processed signal from the computer through a GSM and make the phone call. Mikhael [9] experiments yielded an accuracy of 99.38% word correctness, 98.75% word accuracy and 90.0% sentence correctness by triphones model.

Although, the two researchers mentioned above have identified an important gap in the area of Amharic ASR system for mobile phones and made their contribution to narrow the gap, further research and development works still need to be done to ensure the maximum possible usability of mobile phones by the whole community, as the case with the technologically favored languages.

And because, it is recommended that, future local research works in the area of speech recognition should focus on the development of products that can benefit the community at large, and also because, these days, mobile phones are becoming a very common computing and communication devices, this research work has focused on investigating an improved way of integrating an Amharic speech recognition system with mobile phone devices [11]. Thus, this research work is an extension to Tinbit's [8] and Mikhael's [9] works and has a goal of narrowing the gaps further by improving the limitations of the two works, *i.e.* by improving the recognition accuracy, the case with Tinbit [8]; and by making the recognizer a fully offline/embedded, the case with Mikhael [9] and also by increasing the vocabulary size which is the case with both of the two works.

In light of this, in this study, a total of 36 Amharic words are used to build two acoustic models namely, word-dependent phone (WDP) and word-dependent CV syllable (WDCVS) models. And from the 36 words 21 of them, which in combination can form 101 Amharic digits (digits from 0 to 100) as spoken in a connected manner, are used to train the models for Amharic digits. The remaining 15 words, which can form 14 distinct commands, are used to train the models for commonly used command and control phrases in Amharic language.

In general, this research work has investigated and explored an efficient way of implementing a connected word, offline/embedded, speaker-dependent, small vocabulary Amharic speech recognition system for Mobile phones using the two models, WDP and WDCVS. The mobile phones used to implement the ASR system are, mobile phones installed with Android – a platform which is the most popular and adaptable mobile phone Operating System [12].

1.3 Research Questions

Based on the statement of the problem described above, this research attempted to provide answer to the following questions: -

1. How can a better recognition accuracy be achieved in Amharic ASR system for mobile phones?
2. How can the acceptable sequences of words in the recognizer be modeled efficiently?
3. Which one of the two models, WDP and WDCVS, is the best for building a connected word Amharic ASR that can be used to command and control a mobile phone?
4. What are the challenges in building a connected word Amharic ASR system for mobile phones?

1.4 Objectives of the Study

1.4.1 General Objective

- To explore ways of creating and integrating a connected word, offline, speaker-dependent, small vocabulary Amharic ASR system, that can be used to command and control mobile phones.

1.4.2 Specific Objectives

- To perform a comprehensive literature review related to ASR, ASR on mobile phones and the language under study which is Amharic.
- To identify and select some digit words and common mobile phone command and control functionality phrases, translate them to Amharic, and prepare a speech corpus of these words.
- To design and develop an offline, speaker-dependent, small vocabulary, connected word WDP and WDCVS based Amharic speech recognition system.
- To test the effect the three model types i.e. continuous, semi-continuous and PTM, have on the WDP and WDCVS models and select the WDP and WDCVS models with best recognition accuracies.
- To build a prototype Android application and integrate the developed ASR models.
- To evaluate the usability and accuracy of the prototype application.
- To forward conclusion and recommendation for further study in the area.

1.5 Methodology

In order to achieve the specified objectives, the following methods, approach, techniques and tools are employed during the course of implementation.

1.5.1 Literature Review

Comprehensive literature review of books, journal articles, and relevant documents from the Internet and other sources is performed in order to understand the underlying principles and theories of the various approaches, techniques and tools that can be employed in integrating a speech recognition system on mobile phones.

Literatures related to the language under study which is Amharic are also reviewed to gain a clear understanding of the language since it is fundamental in building a speech recognition system for the language. In addition, literatures related to integrating a speech recognition system on an Android mobile phone for different languages are reviewed. Furthermore, research works done both locally and internationally in the area of speech recognition are investigated as it helps the researcher to have a good understanding of the problem area and to support the results in a scientific manner and evaluate empirically.

1.5.2 Development Method, Tools and Techniques

In the development process of the prototype speech recognizer system, the researcher has used the appropriate tools and techniques indicated in different literatures. Specifically, the researcher employed the Hidden Markov Model (HMM) for developing the acoustic models. The reason why HMM is selected for this experimental research is, its popularity and widespread acceptance [13] and also because it is contained in the Pocketsphinx – the system chosen for this research.

The reason for choosing Pocketsphinx, a tool from CMUSphinx system is because it is an open source speech recognition system which is targeted to be a lightweight speech recognition engine, specifically tuned for handheld and mobile devices, though it works equally well on PCs with the ability of recognizing speech in real-time. It is also trainable to recognize different languages and dialects [14]. In addition to Pocketsphinx, Sphinxtrain – acoustic model training tool, and Sphinxbase – a support library required by Pocketsphinx, both of which are also from CMUSphinx, are used in the development and experimentation phases of this research work.

To deploy the developed recognizer, embedded speech recognition approach is employed. In this approach, all speech recognition processing activity is carried out in the target mobile or handheld consumer device. In other words, an ASR system deployed with this approach can be used without needing to have a network connection, that is to say, at any time and place.

An Android application which acts as an interface between the recognition engine and the user, is created using the latest version, version 2.3.3, of Android Studio. After the application is created, the developed acoustic models are incorporated into the developed application to create the prototype Amharic speech recognition system for mobile phones.

1.5.3 Data Preparation

As mentioned above, a small vocabulary consisting of 36 Amharic words is employed in this research. This is mainly due to the fact that, this amount of words will cover the Amharic digits from 0 to 100 when used in combination and other 11 commonly used phrases in commanding and controlling a mobile phone. In addition, though WDP and WDCVS models are used in this research, these models are equivalent with whole word-based models [15]; and thus, share some characteristics of word based models. Therefore, in this study, it has been found necessary to limit the vocabulary size as it is recommended to limit the number of words when building a word-based speech recognition system [16].

Though, there are many commands in mobile phones that can be carried out by voice, we have used our own observation and experience to select only some of the common commands (which are 9 in number), and to employ in this research. Similarly, we've selected the digit words that can form the Amharic digits from 0 up to 100, because these digits can form a two-digit combination, which is often used in dialing a phone number in mobile phones.

A speech corpus is one of the fundamental requirements of any speech recognition system, and because there does not exist a reference speech corpus that can be readily used in this research work, a speech corpus of the selected words is prepared. The speech corpus is recorded from one person using PC and also a mobile phone using the appropriate software. The recorded data is used for both training and testing the models after dividing it into 82% and 18% for training and for testing respectively.

In addition to preparing the speech corpus, all the components needed in developing a speech recognition system *i.e.* transcription file, pronunciation dictionary, filler dictionary, phone file and grammars are also created. In this study, the pronunciation dictionaries used are, word-dependent phone and word-dependent CV syllable, as shown in Appendix 1-A and Appendix 1-B.

1.5.4 Evaluation and Testing Procedures

As clearly indicated in different literatures, testing activities are one of the important tasks to be performed while doing an experimental research. The standard evaluation metric for speech recognition systems is the word error rate (WER) [17]. Hence, in this study, two categories of

performance evaluations, Category I and Category II, are conducted and WER is used in both categories of the performance evaluations.

The first category of performance evaluation, Category I, is aimed at determining the effect that the three model types CMUSphinx support: continuous, semi-continuous and PTM, have on the two models used in this research work namely, WDP and WDCVS. Accordingly, a total of six distinct performance evaluations are conducted, after training each of the two models with the three model types. These performance evaluations are carried out objectively using the test set by executing the appropriate commands on each of the trained models.

The second category of performance evaluation, Category II, is conducted on a mobile phone by incorporating each of the trained models into the developed application, and then by installing the prototype application on a mobile phone. After the prototype application is installed, the evaluation is conducted subjectively by speaking all the acceptable utterances to the prototype application. This performance evaluation is thus, used to determine which of the WDP and WDCVS models trained with the three model types have scored better recognition performance on a mobile phone, and also to decide whether the WDP or WDCVS model is the best.

1.6 Significance of the Study

This study will have its own contribution in supporting the efforts in making technology accessible and convenient for the whole community. Since speech can be used to control electronic devices like computer and smartphones by verbalizing text or commands, it offers convenience to users such as those who are physically incapacitated, or those who suffer from tedious strain injury. They can be relieved from having to worry about typing, touching or working in general with their mobile phones since it allows them to control and command their mobile phone devices merely by using their voice.

In addition, this study will have a great role in promoting researches and developments related to Amharic speech recognition for mobile phones.

Moreover, this study will also have a considerable importance in filling the literature gaps associated to this specific research area, which was a big challenge while conducting this study; particularly, the gaps related to the use of word-dependent phone models in CMUSphinx speech

recognition toolkit and also the literature gaps related to incorporating an offline ASR system with a mobile phone.

1.7 Scope and limitations of the study

This research aims at investigating and exploring an enhanced way of integrating an automatic speech recognition system that can be used in commanding and controlling mobile phone devices using Amharic language. For this, a connected word, offline, speaker-dependent, small vocabulary Amharic speech recognition system is developed and therefore, all the other features of speech recognition system are not part of this study.

This research work also does not consider the dialect differences in the Amharic language, mainly because it needs performing a lot of acoustic tasks which could be beyond the financial and time restrictions of this research work. Likewise, pronunciation variations of some words are also not considered.

Currently, even though there are different kinds of mobile phones – from different manufactures with different operating systems, that are common in both the market and the community, this research work has only investigated the integration of an Amharic speech recognition system on mobile phones shipped with an Android operating system.

1.8 Organization of the thesis

This thesis is organized in six chapters and the contents of each chapter are described as follows. The first chapter which is chapter one introduces the whole paper and is consisted of the sub-topics: background of the study, statement of the problems and its justification, objectives of the research, research questions, methodology, significance of the study and finally scope and limitation of the study.

The second chapter, which is review of related literatures, discusses the different underlying principles and theories in speech technologies. It starts by discussing the speech recognition systems and then discusses the Hidden Markov Model (HMM). Continuing, this chapter presents the Amharic Language, related speech recognition on mobile phones and finally, previous related works in the area of this particular research work are presented.

In the third chapter, the approaches, architecture of the system and also the tools and techniques used in the development of the prototype Amharic speech recognizer for mobile phones are discussed in detail.

The fourth chapter, is all about the experimentation and evaluation carried out on the developed models. And therefore, in this chapter, the researcher tried to show the results of the experimentations in line with a discussion on the results obtained from the developed prototype application.

The last chapter, which is chapter five, presents the conclusions arrived in by the researcher and also the recommendations that need to be considered in future works in the area.

Chapter Two

Literature Review

2.1 Introduction

This chapter presents an overview discussion of the speech recognition system found from different related literatures. The first section presents definition, classifications and architecture of the generic speech recognition system. Next, the Hidden Markov Model is discussed briefly. Following, although speech recognition system on Mobile phone is almost similar in many ways with that of the speech recognition on computers, there are some issues that are specific to speech recognition on mobile phones, and these issues like the architectural designs of speech recognition and speech recognition challenges in the mobile context are discussed in detail. Finally, previous related research works, for both Amharic and other languages are presented.

2.2 Automatic Speech Recognition

Speech Recognition is one of the thrust research areas in speech processing and is also known as Automatic Speech Recognition (ASR). It is a process of converting a speech signal to a sequence of words (*i.e.*, spoken words to text) by means of an algorithm implemented as a computer program [16].

A source-channel mathematical model or a type of generative statistical model is often used to formulate speech- recognition problems. As illustrated in Figure 2.1, the speaker's mind decides the source word sequence W that is delivered through his or her text generator. The source is passed through a noisy communication channel that consists of the speaker's vocal apparatus to produce the speech waveform and the speech signal-processing component of the speech recognizer. Finally, the speech decoder aims to decode the acoustic signal X into a word sequence $\widehat{W}$, which is in ideal cases close to the original word sequence W [18] [16].

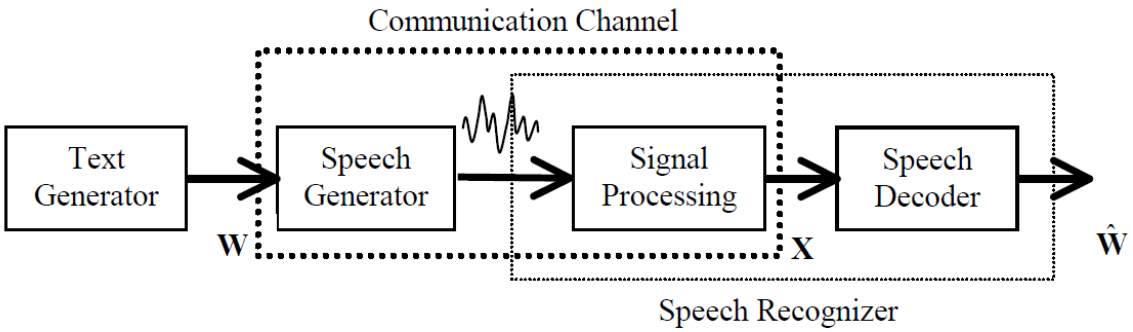


Figure 2.1 A source-channel model for a speech recognition system (extracted from [19])

Modern speech recognition systems have been built invariably based on statistical principles, as pioneered by the work of [20] and [21] and exposed in detail in [16] as cited in [18]. The work of speech recognition can be succinctly described as follows by the fundamental equation of statistical speech recognition:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{w}} P(\mathbf{W}|\mathbf{A}) = \arg\max_{\mathbf{w}} \frac{P(\mathbf{W})P(\mathbf{A}|\mathbf{W})}{P(\mathbf{A})} \qquad (2.1)$$

where for the given acoustic observation or feature vector sequence $X = X_1X_2 \ldots X_n$, the goal of speech recognition is to find out the corresponding word sequence $\hat{W} = w_1w_2 \ldots w_m$ that has the maximum posterior probability $P(W|X)$ as expressed with Equation 2.1. Since the maximization of Equation 2.1 is carried out with the observation X fixed, the above maximization is equivalent of the maximization of the numerator:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{w}} P(\mathbf{W})P(\mathbf{X}|\mathbf{W}) \qquad (2.2)$$

where P(W) and P(X|W) constitute the probabilistic quantities computed by the language modeling and acoustic modeling components, respectively, of speech-recognition systems.

The practical challenge is how to build accurate acoustic models, P(X|W), and language models, P(W), which can truly reflect the spoken language to be recognized. For large vocabulary speech recognition, we need to decompose a word into a sub word sequence (often called pronunciation modeling), since there are a large number of words. Thus, P(X|W) is closely related to phonetic

modeling. P(X|W) should take into account speaker variations, pronunciation variations, environmental variations, and context-dependent phonetic co-articulation variations. Last, but not least, any static acoustic or language model will not meet the needs of real applications. So it is vital to dynamically adapt both P(W) and P(X|W) to maximize P(W|X) while using the spoken language systems. The decoding process of finding the best-matched word sequence, W, to match the input speech signal, X, in speech-recognition systems is more than a simple pattern recognition problem, since one faces a practically infinite number of word patterns to search in continuous speech recognition [18].

2.3 Classification of Speech Recognition Systems

Different scholars classify Speech Recognition Systems into different classes based on some parameters that are related to the task of speech recognition. Some of the classification based on parameters are:

2.3.1 Classification based on Utterances

According to Saksamudre et. al. [22], speech recognition systems can be separated in different classes based on what types of utterances they have the ability to recognize. Accordingly,

2.3.1.1 Isolated Words

These systems usually require each utterance to have quiet (lack of an audio signal) on both sides of the sample window. It accepts single words or single utterance at a time. Isolated word recognition is suitable for situations where the user is required to give only one word response or command. It is simple and easy for implementation because word boundaries are obvious and the words tend to be clearly pronounced which is the major advantage of this type.

2.3.1.2 Connected Words

Connected word systems (or more correctly 'connected utterances') are similar to isolated words, but allows separate utterances to be 'run-together' with a minimal pause between them [22]. The primary goal of connected word ASR systems is to create an ASR system capable of recognizing a fluently spoken string of words (e.g., digits) based on matching a concatenated pattern of individual words [1].

2.3.1.3 Continuous Speech

Continuous speech recognizers allow users to speak almost naturally, while the computer determines the content. Basically, it's computer dictation. Recognizers with continuous speech capabilities are some of the most difficult to create because they utilize special methods to determine utterance boundaries.

2.3.1.4 Spontaneous Speech

At a basic level, it can be thought of as speech that is natural sounding and not rehearsed. An ASR system with spontaneous speech ability should be able to handle a variety of natural speech features such as words being run together with "ums" and "ahs", and even slight stutters.

2.3.2 Classification based on Speaker Model

Each speaker has unique voice due to several reasons *i.e.* physical body, health condition, personality, age and etc. Thus, according to Saksamudre et. al., [22], based on speaker model, speech recognition systems can be classified into three main categories as follows:

2.3.2.1 Speaker Dependent Models

Speaker dependent systems are developed for a particular type of speaker. They are generally more accurate for the particular speaker, but could be less accurate for other type of speakers. These systems are usually cheaper, easier to develop and more accurate. But these systems are not flexible as that of speaker independent systems.

2.3.2.2 Speaker Independent Models

Speaker Independent system can recognize a variety of speakers without any prior training. A speaker independent system is developed to operate for any particular type of speaker. It can be used in systems that must accept input from a large number of different users. Its drawback is that it limits the number of words in a vocabulary. Implementation of Speaker Independent system is the difficult and also it is expensive and its accuracy is lower than that of speaker dependent systems.

2.3.2.3 Speaker Adaptive Models

Speaker adaptive speech recognition system uses the speaker dependent data and adapt to the best suited speaker to recognize the speech and decreases error rate by adaption. They adapt operation according to characteristics of speakers.

2.3.3 Classification based on Vocabulary

Another classification of speech recognition systems is based on vocabulary size. S. K. Saksamudre et. al., [22] state That that size of vocabulary of speech recognition system can affect the complexity, processing and the rate of recognition of the speech recognition system. Based on the vocabulary size, [22] classify ASR systems as:

- Small Vocabulary - 1 to 100 words or sentences
- Medium Vocabulary - 101 to 1000 words or sentences
- Large Vocabulary- 1001 to 10,000 words or sentences
- Very-large vocabulary - More than 10,000 words or sentences

2.4 Speech Recognition Architecture

A typical speech recognition system is developed with major components that include acoustic front-end, acoustic model, lexicon, language model and decoder as shown in Figure 2.2. Acoustic front-end takes care of converting the speech signal into appropriate features which provides useful information for recognition. The input audio waveform from a microphone is converted into a sequence of fixed-size acoustic vectors is a process called feature extraction. The parameters of word or sub-word models are estimated from the acoustic vectors of the training data. The decoder operates by searching through all possible word sequences to find the sequence of words that is most likely to generate. The likelihood is defined by an acoustic model $P(O|W)$ and $P(W)$ is determined by a language model [23].

The functionality of automatic speech recognition system can be described as an extraction of a number of speech parameters from the acoustic speech signal for each word or sub-word unit. The speech parameters describe the word or sub-word by their variation over time and together they build up a pattern that characterizes the word or sub-word. In a training phase the operator will read all the words of the vocabulary of the current application. The word patterns are stored and later when a word is to be recognized its pattern is compared to the stored patterns and the word

that gives the best match is selected. This technique is generally referred to as pattern recognition [16] [23].

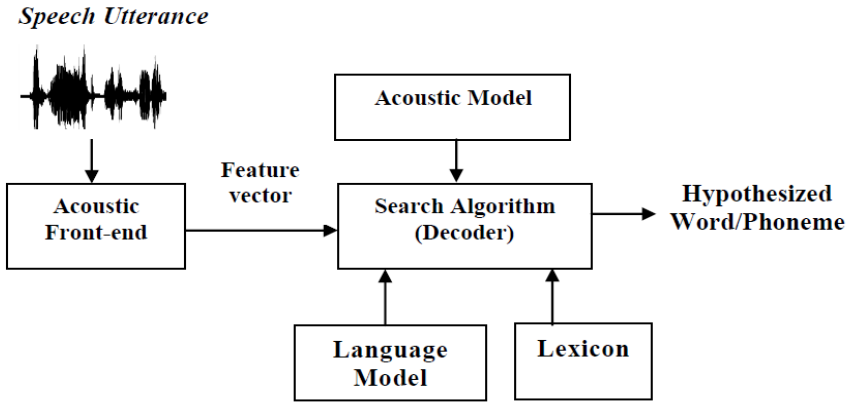


Figure 2.2 Speech Recognition Architecture (extracted from [23])

2.4.1 Acoustic Front-end

Acoustic front-end involves signal processing and feature extraction. In speech recognition, the main goal of the feature extraction step is to compute a parsimonious sequence of feature vectors providing a compact representation of the given input signal [24].

The feature extraction is usually performed in three stages. The first stage is called the speech analysis or the acoustic front end. It performs some kind of spectra temporal analysis of the 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 (which is not always present) transforms these extended feature vectors into more compact and robust vectors that are then supplied to the recognizer [23].

In order to find some statistically relevant information from incoming data, it is important to have mechanisms for reducing the information of each segment in the audio signal into a relatively small number of parameters, or features. These features should describe each segment in such a characteristic way that other similar segments can be grouped together by comparing their features. There are many feature representations in use, but the most common is the mel-frequency cepstral coefficient (MFCC) feature set [16].

2.4.2 Acoustic Model

Acoustic model is one of the most important knowledge sources for automatic speech recognition system, which represents acoustic features for phonetic units to be recognized. In building an acoustic model, one fundamental and important issue is choosing of basic modeling units. Generally speaking, when the target language of the speech is specified, there is several types of sub word unit can be used for acoustic modeling. Different acoustic modeling unit can make a dramatic difference on the performance of the speech recognition system [23].

Acoustic modeling of speech typically refers to the process of establishing statistical representations for the feature vector sequences computed from the speech waveform. Although there are many acoustic models, Hidden Markov Model (HMM) is one of the most commonly used statistical models to build acoustic models. An acoustic model is a file that contains statistical representations of each of the distinct sounds that makes up a word. Each of these statistical representations is assigned a label called a phoneme acoustic model is created by taking a large database of speech called a speech corpus and using special training algorithms to create statistical representations for each phoneme in a language. Each phoneme has its own HMM. The speech decoder listens for the distinct sounds spoken by a user and then looks for a matching HMM in the acoustic model. Each spoken word is decomposed into a sequence of basic sounds called base phones. The acoustic model describes the probability of a specific observation given a base phone [23] [22].

2.4.3 Language Model

A language model is a collection of constraints on the sequence of words acceptable in a given language. These constraints can be represented, for example, by the rules of a generative grammar or simply by statistics on each word pair estimated on a training corpus. Although there are words that have similar sounding phone, humans generally do not find it difficult to recognize the word. This is mainly because they know the context, and also have a fairly good idea about what words or phrases can occur in the context. Providing this context to a speech recognition system is the purpose of language model. The language model specifies what are the valid words in the language and in what sequence they can occur [23].

Language models are usually trained, that is, the n-gram probabilities are estimated by observing sequences of words in corpora of text that contain, typically, millions of word tokens and by reducing perplexity on training data [17].

It has been observed however that reduced perplexity does not necessarily lead to better speech recognition results. Therefore, algorithms that improve language models based on their effect on speech recognition are particularly appealing a language model that specifies the probability distribution of words the speaker may utter next, given a history of uttered words. Common language models are bigram and trigram models. These models contain computed probabilities of groupings of two or three particular words in a sequence, respectively. There are tools for language modeling like CMU Statistical Language Modeling (SLM) Toolkit, Stanford Research Institute Language Modeling Toolkit [23].

2.4.4 Decoder

In the decoding stage, the task is to find the most likely word sequence W given the observation sequence O, and the acoustic-phonetic-language model. The decoding problem can be solved using dynamic programming algorithms. Rather than evaluating likelihoods of all possible model paths generating O, the focus is on finding a single path through the network yielding the best match to O. To estimate the best state sequence for the given observation sequence, the Viterbi algorithm is frequently used [1].

In the case of larger vocabulary recognition tasks, it would be challenging to consider all possible words during the recursive part of the Viterbi algorithm. To address this, a beam search can be used for Viterbi iteration, only the words with path probabilities above a threshold are considered when extending the paths to the next time step. This approach speeds up the searching process at the expense of decoding accuracy. The Viterbi algorithm assumes that each of the best paths at time t must be an extension of each of the best paths ending at time $t - 1$, which is not generally true. The path that seems to be less probable than others in the beginning may turn into being the best path for the sequence as a whole (e.g., the most probable phoneme sequence does not need to correspond to the most probable word sequence). This issue is addressed by extended Viterbi and forward-backward algorithms [17].

2.5 Recognition units

One of the important issues in designing a speech recognition system is the selection of appropriate modeling unit for a recognition task. Different recognition units may be preferable in different settings, such as high-variability conversational speech, high-noise conditions, low-resource settings, or multilingual speech recognition. There are several recognition unit exist, such as words, phoneme, di-phone, tri-phone, senone, syllable, demi-syllable, acoustic unit, morpheme, grapheme [25].

According to Huang, Acero and Hon [16], the following criteria need to be considered when choosing an appropriate modeling unit:

- The unit should be *accurate* – to represent the acoustic realization that appears in different contexts.
- The unit should be *trainable* – we should have enough data to estimate the parameters of the unit.
- The unit should be *generalizable* – so that any new word can be derived from a predefined unit inventory for task-independent speech recognition.

From the various recognition units, phoneme, syllable and words are the commonly used recognition units in many speech recognition tasks.

2.5.1 Words

As the most natural unit of speech, whole-word models have been widely used for many speech recognition systems. A distinctive advantage of using word models is that we can capture phonetic co-articulation inherent within these words. When the vocabulary is small, we can create word models that are context dependent. For example, if the vocabulary is Amharic digits from "ዜሮ" to "ዘጠኝ", we can have different word models for the word "አንድ", to represent the word in different contexts. Thus, each word model is dependent on its left and right context. If someone says "ሦስት", "አንድ", "ሁለት", the recognizer uses the word model "አንድ" that specifically depends on the left context "ሦስት" and right context "ሁለት". Since the vocabulary is small (10), we need to have only 10*10*10 = 1000 word models, which is achievable when you collect enough training data. With context-dependent, or even context independent, word models, a wide range of phonological variations can be automatically accommodated. When these word models are adequately trained,

they usually yield the best recognition performance in comparison to other modeling units. Therefore, for small vocabulary recognition, whole-word models are widely used, since they are both accurate and trainable, and there is no need to be generalizable. Though, words are suitable units for small-vocabulary speech recognition, they are not a practical choice for large-vocabulary continuous speech recognition due to the three facts mentioned above [16].

2.5.2 Phones

Most commonly used sub-word unit in speech recognition is the phone – a basic speech sound such as a single consonant or vowel. Each word is then represented as a sequence, or several alternative sequences of phones specified in a pronunciation dictionary [25]. Unlike word models, phonetic models provide no training problem as there is a significantly less number of phones in any given language as compared to the number of words. Moreover, they are also vocabulary independent by nature and can be trained on one task and tested on another. Thus, phones are more trainable and also are generalizable.

However, the phonetic model is inadequate because it assumes that a phoneme in any context is identical. Although we may try to say each word as a concatenated sequence of independent phonemes, these phonemes are not produced independently, because our articulators cannot move instantaneously from one position to another. Thus, the realization of a phoneme is strongly affected by its immediately neighboring phonemes. For example, if context-independent phonetic models are used, the same model for t must capture various events, such as flapping, unreleased stops, and realizations in /t s/ and /t r/. Then, if /t s/ is the only context in which t occurs in the training, while /t r/ is the only context in the testing, the model used is highly inappropriate. While word models are not generalizable, phonetic models overgeneralize and, thus, lead to less accurate models. This, problem can be efficiently handled by making the phonetic models context dependent, provided there are enough training data to estimate these context-dependent parameters. Context-dependent phonemes have been widely used for large-vocabulary speech recognition, thanks to its significantly improved accuracy and trainability. A context usually refers to the immediately left and/or right neighboring phones [16].

2.5.3 Word-dependent phones(WDP)

Word-dependent phones compromise between the two models discussed above, which are word and phone models. Word-dependent phone models use phones; however, these phones are word-dependent. That is to say, a phone model used for one word has different parameters from the model of the same phone in another word; or in other words, in this model, a phone used in two different words will have two different parameters. Like word models, word-dependent phone models can model word-dependent phonological variations, but require considerable training and storage [26].

WDP modeling is better than word modeling in two ways. The first is, when a word has not been observed frequently, its parameters can be interpolated (or averaged) with that of context-independent phone models. Second, a new word need not be repeated many times – context-independent phone models can be used to get acceptable performance and this holds true for a word in the vocabulary but not observed in the training data [26].

WDP modeling was proposed by Chow et al. [27] as cited in [26], and in their study Chow et al. interpolated WDP models with context-independent phone models using empirically determined weights. Their experimental results yielded error rates of 10% for the WDP models, 14% for word models, and an error rate of 24% for phone models. Thus, the results clearly shows that the WDP models actually outperformed the two other models.

2.5.4 Syllables

Syllable is usually a larger unit than a phone, since it may encompass two or more phonemes and therefore, it can be taken as a compromise between the word and phonetic model. There are a few cases where a syllable may only consist of single phoneme. Syllables are the phonological building blocks of words. Syllables have a vital role in a language's rhythm, prosody, poetic meter and stress. The syllable as a unit, naturally accounts for the severe contextual effects among its phones as in the case of words. It accounts for pronunciation variation more systematically than a phone. Syllables are longer and less context sensitive than phones and capable of exploiting both the spectral and temporal characteristics of continuous speech. Moreover, the syllable has a close connection to articulation, integrates some co-articulation phenomena, and has the potential for compact representation of conversational speech. Many ASR work carried out with syllable based models in English, Chinese and Amharic. Japanese there are only 100 distinct syllables and 233

distinct CV syllables in Amharic compared to English where number of distinct syllables are 30,000. When there is large number of syllables, it becomes difficult to train syllable models for ASR [16] [25] [28].

2.6. Hidden Markov Model (HMM)

2.6.1 Definition

The hidden Markov model (HMM) is a very powerful statistical method of characterizing the observed data samples of a discrete-time series. Not only can it provide an efficient way to build parsimonious parametric models, but can also incorporate the dynamic programming principle in its core for a unified pattern segmentation and pattern classification of time-varying data sequences. The data samples in the time series can be discretely or continuously distributed; they can be scalars or vectors [16].

According to L. R. Rabiner [29], hidden Markov model is a doubly embedded stochastic process with an underlying stochastic process that is not directly observable (it is hidden) but can be observed only through another set of stochastic processes that produce the sequence of observations. Below, the mathematical formulation of Hidden Markov Models is presented as it is crucial in building an HMM based speech recognition. The notations will be done to remain in the contexts cited in [29].

Consider a finite set of states which may be described at any time as being in one of a set of N distinct states s1, s2, ..., sN. Figure 2.3 depicts the hypothetical example for five states HMM that at regularly spaced discrete times, undergoes a change of state (possibly back to the same state) according to a set of probabilities associated with the state.

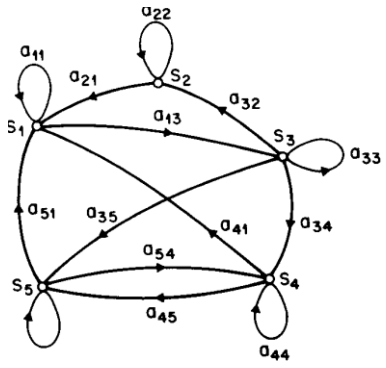


Figure 2.3 A Markov chain with 5 states (extracted from [13])

2.6.2 Elements of HMM

1) N, Number of hidden states in the model: although the states are hidden, for many practical applications there are some physical significance attached to the states or to the set of the states of the model. In general, the states are interconnected in such a way that any state can be reached from any other state. Here, we denote the individual states as:

$S = \{S_1, S_2, \ldots, S_N\}$

2) M, number of distinct observation symbols per states: the observation symbols correspond to the physical output of the system being modeled. Therefore, we denote the individual symbols as:

$V = \{V_1, V_2, \ldots, V_M\}$

3) The state transition probability distribution $A = \{a_{ij}\}$ where,

$Aij = P[q_{t+1} = S_j | q_t = S_i] \qquad 1 \leq i, j \leq N.$

For the special case, where any state can reach any other state in a single step, we have $a_{ij} > 0$ for all i, j. For other types of HMMs, $a_{ij} = 0$ for one or more (i, j) pairs.

4) The observation symbol probability distribution in state j, $B = \{b_i(k)\}$, where

$$b_j(k) = P[V_k \text{ at } t | q_t = S_j], \qquad 1 \leq j \leq N$$
$$1 \leq k \leq M$$

5. The initial state distribution $\pi = \{\pi_i\}$ where

$$\pi_i = P[q_1 = S_i], \qquad 1 \leq i \leq N.$$

Given appropriate values of N, M, A, B and π, the HMM can be used as a generator to give an observation sequence

$$O = O_1 O_2 \ldots . O_T$$

(Where each of the observation O_t is one of the symbols from V and T is the number of observations in the sequence) as follows;

a. choose an initial state $q_1 = S_i$ according to the initial state distribution π

b. set $t = 1$.

c. choose $O_t = V_k$ according to the symbol distribution in state Si, that is, $b_i(k)$.

d. transit to new state $q_{t=1} = S_j$ according to the state transition probability distribution for state Si, that is, a_{ij}.

e. set $t = t + 1$ and return to step c, if $t < T$; otherwise terminate the procedure.

It can be seen from the above discussion that a complete specification of the HMM requires, specification of two model parameters (N and M), specification of observation symbol and the specification of the three probability measure A, B and π. Therefore, the entire model is represented as;

$\lambda = (A, B, \pi)$

2.6.3 The Basic Problems for HMMs

According to R. L. Rabiner [29], given the above fact and discussions of the HMM, there are three basic problems of interest that must be solved for the model to be useful in real-world applications. These problems are the following:

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

Given a model and a sequence of observations, how do we compute the probability that the observed sequence was produced by the model or how well a given model matches a given observation sequence, that is, an evaluation and a scoring for how well a given HMM matches a given observation sequence.

Problem 2: Given the observation sequence $O = O_1 O_2 \ldots OT$ and the model λ, how do we choose a corresponding state sequence $Q = q_1 q_2 \ldots q_T$ which is optimal in some meaningful sense?

Here, we attempt to uncover the hidden parts of the model or find the correct state sequences. In case of practical situation, we search for optimality criteria to solve those problems. Typical uses might be to learn the structure of the model, to find optimal state sequence for continuous speech recognition.

Problem 3: How do we adjust the model parameters l = (A, B,p) to maximize P(O|l)?

Here, the concern is optimizing the model parameters so as to best describe how a given observation sequence comes about. In other words, problem 3 refers to the learning or training the HMM to optimally adapt model parameters to observed training data.

2.7 Amharic Language

Amharic was the national language of Ethiopia until 1991 E.C. Currently, it is the official working language of the Federal Democratic Republic of Ethiopia and thus has official status nationwide and it is the official or working language of several regional states/regions, with the exceptions of Tigray, Harari, Somali and Oromia. It was also, for a long period, the principal literal language and medium of instruction in primary schools in the country, while secondary and higher education is carried out in English. Throughout the country Amharic is taught as a subject in schools. Many universities have also departments of Amharic to train Amharic teachers, journalists, and people who work in public relations. Currently, different mass medias like radios, television broadcasts and the press in the country use Amharic for disseminating information to the public [30].

2.7.1 The Phonology of Amharic

Phonetics is the study of speech sounds and their production, classification, and transcription. Phonology on the other hand, is the study of the distribution and patterning of speech sounds in a language and of the tacit rules governing pronunciation [16]. Articulatory phonetics deals with how the human vocal apparatus is manipulated to produce sounds. The basic assumption of articulatory phonetics is that sounds are best described in terms of the configurations of the vocal tract necessary to utter the sounds. Sounds can also be classified as vowels and consonants which are produced in different ways. While consonants are articulated with a substantial degree of obstruction in the oral cavity, vowels are produced with a relatively free airflow [31].

2.7.1.1 Amharic Consonants

According to Voicing, Manner and Place of articulation Amharic consonants are described as follows.

Manner of articulation	Point of Articulation	Bilabial	Labio-dental	Alveolar	Palatal	Velar	Labio-Velar	Glottal
Stops	Voiced	⠃ /b/		⠙ /d/		⠛ /g/	⠣ /g^{w}/	
	Voiceless	⠏ /p/		⠞ /t/		⠅ /k/	⠩ /k^{w}/	⠀ (?)
	Ejective	⠨ /p’/		⠠ /t’/		⠫ /k’/	⠭ /k’^{w}/	
Fricatives	Voiced		⠧ /v/	⠵ /z/	⠾ /ž/			
	Voiceless		⠋ /f/	⠎ /s/	⠷ /š/			⠓ /h/⠭ /h^{w}/
	Ejective			⠸ /s’/				
Affricates	Voiced				⠚ /j/			
	Voiceless				⠦ /č/			
	Ejective				⠳ /c’/			
Nasals		⠍ /m/		⠝ /n/	⠻ /ñ/			
Liquids				⠇ /l/	⠗ /r/			
Glides		⠺ /w/			⠽ /y/			

Table 2.1 Categories of Amharic Consonants (adapted from [32])

2.7.1.2 Amharic Vowels

Vowels are open sounds, made largely by shaping the vocal tract rather than by interfering with the flow of air stream. Vowels are most usefully described in terms of the position of the tongue as they are articulated. A vowel articulated with the body of the tongue relatively forward is classified as a *front vowel*; one made with the body of the tongue relatively high is a *high vowel*. Vowels produced with the body of tongue neither high nor low are called mid vowels. Vowels produced with the tongue body front are called front vowels while those made with the tongue body back are called *back vowels.* Those vowels made with the tongue body neither front nor back are called central vowels. Vowels accompanied by lip rounding as in (u and o) are called rounded vowels while the other vowels are called unrounded vowels. Amharic has 7 (seven) vowels their description is given in the table below [31].

	Front/Unrounded	Central/Unrounded	Back/Rounded
High	⠊ /i/	⠁ /I/	⠥ /u/
Mid	⠑ /e/	⠯ /E/	⠕ /o/
Low		⠁ /a/	

Table 2.2. Amharic Vowel Articulations (adapted from [32])

2.7.2 The Amharic Writing System

Written Amharic uses a unique script called Fidel which has originated from the Ge'ez alphabet (liturgical language of the Ethiopian Orthodox Church). The oldest Amharic inscription was derived from the Sabean writing which has had twenty-seven symbols in its unvocalized shape. But later Geez pursued the most original course taken Semitic script in denoting vowels by a variety of changes in the structure of the consonantal symbol. Vowels have thus become an integral part of Amharic writing which now assumed the character of a syllabary – consonant-vowel (CV) phoneme pairs. The script also has a unique set of punctuation marks and digits, and some special characters for labialized consonants [33].

Undergoing many transformations through the ages, the Amharic script has now 33 core characters each of which occurs in seven orders (one basic form and six non-basic forms) (as shown in Appendix 2). The seven orders represent syllable combinations consisting of a consonant and following vowel sounds (ኧ ኡ ኢ ኣ ኤ እ ኦ). That is each of the 33 Amharic characters has seven forms representing a consonant and a vowel at the same time which makes the Amharic script syllabic. The first order is the basic form and there are 33 basic forms giving 231 characters [34] cited in [35]. Other symbols representing labialization, numerals, and punctuation marks are also available. These bring the total number of Amharic scripts to 310.

As an example, the symbolic representations of the seven forms of the Amharic characters በ(be) and ገ(ge) are shown in Table 2.1 below.

Consonant	1st order	2nd order	3rd order	4th order	5th order	6th order	7th order	Description
በ	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ	the seven forms of በ
	ብኧ	ብኡ	ብኢ	ብኣ	ብኤ	ብ	ብኦ	Consonant-vowel representation
	Bä	bu	bi	Ba	be	B	bo	Represented sound
ገ	ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጐ	the seven forms of ገ
	ግኧ	ግኡ	ግኢ	ግኣ	ግኤ	ግ	ግኦ	consonant-vowel representation
	Gä	gu	Gi	Ga	Ge	G	go	Represented sound

Table 2.3 Seven forms of Amharic Characters Consonant (adopted from [36])

Solomon and Menzel [28] state that, speech recognition should only consider distinct sounds instead of all the orthographic symbols, unless there is a need to develop a dictation machine that

includes all of the orthographic symbols. Accordingly, in this research work, redundant orthographic symbols that represent the same syllabic sounds are not used.

2.7.3 The Numerals

The Numerals in Amharic writing system are of two kinds, Cardinals and Ordinal. The Cardinals specify the number of things which are the subject of speech: The Ordinals exhibit the order in which they occur [37]. The Cardinal Numbers in the Amharic are shown as follows:

Arabic	Alpha-numeric	English	Arabic	Alpha-numeric	English
1	አንድ	One	20	ሃያ	Twenty
2	ሁለት	Two	30	ሰላሳ	Thirty
3	ሦስት	Three	40	አርባ	Forty
4	አራት	Four	50	አምሳ/ሀምሳ	Fifty
5	አምስት	Five	60	ስልሳ/ስድሳ	Sixty
6	ስድስት	Six	70	ሰባ	Seventy
7	ሰባት	Seven	80	ሰማኒያ	Eighty
8	ስምንት	Eight	90	ዘጠና	Ninety
9	ዘጠኝ	Nine	100	መቶ	Hundred
10	አስር	Ten	1000	ሺ/ሽህ	

Table 2.4 Amharic Cardinals (adopted from [37])

In Amharic compound cardinals are formed by connecting multiple cardinals as presented below:

Arabic	Alpha-numeric	English
11	አስራ አንድ	Eleven
12	አስራ ሁለት	Twelve
13	አስራ ሦስት	Thirteen
14	አስራ አራት	Fourteen
15	አስራ አምስት	Fifteen
16	አስራ ስድስት	Sixteen
17	አስራ ሰባት	Seventeen
18	አስራ ስምንት	Eighteen
19	አስራ ዘጠኝ	Nineteen

Table 2.5 Amharic Compound Cardinals (adopted from [37])

And the same order is observed with all other digits of Amharic; e.g. “ሃያ አንድ”, “ሰላሳ ሁለት”, “አርባ ሦስት” and etc.

Amharic writing system does not have a symbol for zero, negative, decimal point, and mathematical operators, and because of these the Hindu-Arabic numerals and Latin mathematical operators are used for computational purpose [35].

2.8 Speech Recognition on Mobile Phones

Mobile devices are designed to be carried and used while in motion or during pauses at unspecified locations. With significant technology advances in recent years, mobile devices of today are multi-functional – capable of supporting a wide range of applications for both business and consumer use. For example, PDAs and smart phones provide the user not only with the functionality of making phone calls and sending SMS but also the access to the Internet for e-mail, instant messaging and Web browsing. Mobile devices are now functioning as an extension to the PC to enable people to work on the road or away from office and home [38].

In this ubiquitous computing environment, the use of keypad, stylus and small screen sometimes becomes inconvenient, and speech-centric user interface is foreseen to be a desirable interaction paradigm where automatic speech recognition is the enabling technology [5]. This has led to the growing interest in deploying speech recognition on mobile devices.

As ASR technology has been optimized primarily for general computers in a centralized architecture, specific care is required when incorporating the technology into mobile devices and communication networks, both of which place significant constraints on the use of ASR to its full potential.

According to H. Tan and Lindberg [3] the partition between the ASR components is sharp, enabling flexible architectures when deploying it on the device and in the network. Speech is always captured in the client and the application can reside either in the client or in the server. The decision on where to place the remaining ASR components distinguishes the three approaches: NSR, DSR and ESR, as shown in the bottom panel of Fig. 3.1. The choice of approaches is driven by a number of factors including complexity of components, resources available on the device and in the network, and location of the application.

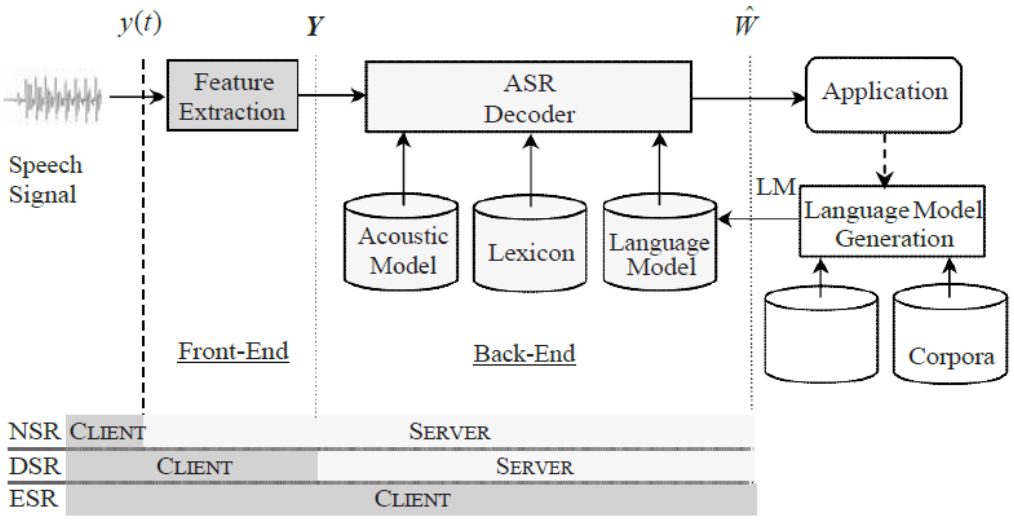


Figure 2.4 Architecture of an ASR system [3].

2.8.1 The Three Architectures and Techniques

Due to the challenges presented in the previous section, speech recognition on mobile devices is deployed with different architectures and the techniques to deal with the hindrances of each architecture is discussed below. The three approaches introduced are: network speech recognition (NSR), distributed speech recognition (DSR) and embedded speech recognition (ESR) [3].

While ESR embeds the whole speech recognition system into the device, NSR and DSR make use of the connectivity available in the device and submit the entire speech recognition processing or part of it to a server. The three architectures are discussed in detail below.

2.8.1.1 Network Speech Recognition

Network Speech Recognition off-loads all recognition related tasks to a network-based server. Speech signals are in most cases encoded by a mobile or Voice-over-IP speech coder and transmitted to the server where feature extraction and recognition decoding are conducted as shown in Figure. 3.4. NSR in its basic form is a concatenation of two systems: a speech encoding-decoding system and a speech recognition system. This enables a plug and play of ASR systems at the server side while no changes are required for the existing devices and networks. The NSR approach has the advantage that numerous commercial applications are developed on the basis of speech coding. It further shares all the advantages of server based solutions in terms of system maintenance and update and device requirements [38].

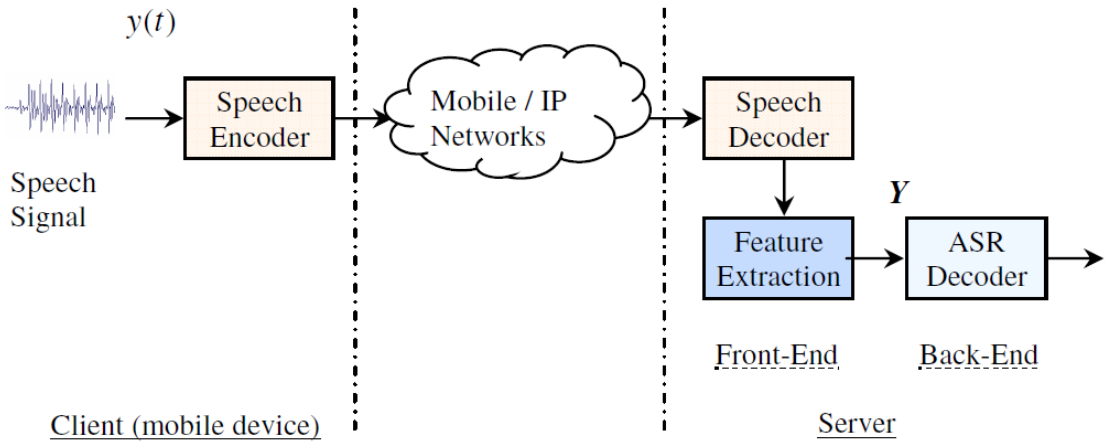


Figure 2.5 Architecture of a network speech recognition system [38].

The disadvantages of NSR are network dependency and distortion introduced by speech transmission specifically by low bit-rate coding and error-prone channels. Coding distortion occurs mainly since speech coders are optimized for receiver-side reconstruction and human listening rather than for computer recognition. For instance, parameterization of speech coding is mainly based on a speech production model and thus the use of linear prediction coding (LPC) coefficients while speech recognition widely employs Mel-frequency cepstral coefficients (MFCCs) that are extracted on the basis of human perception. This difference can be overcome by directly estimating features from the bitstream of coded speech without re-constructing the speech [3].

The effect of packet loss on NSR has been extensively investigated in Mayorga P. et. al. [39] study. The authors reveal that packet loss may imply substantial degradation of recognition performance. In contrast, speech coding is a less severe problem, but when coupled with packet losses, it can make ASR out of function. One of the reasons is that speech coders usually exploit inter-frame correlation to achieve high compression ratio so that one frame loss affects subsequent frames – the phenomenon of error propagation [40].

2.7.1.2 Distributed Speech Recognition

The high complexity of an ASR decoder makes it tempting to adopt a client-server architecture: placing the front-end in the client and the computation-intensive back- end in the server. Since feature extraction is located in the client, the process of speech coding and decoding is eliminated. Instead, the feature vectors are directly compressed and sent to the server for recognition decoding.

As data transmission may take place via heterogeneous networks, the use of a DSR codec further avoids the problem of transcoding [3].

To optimize DSR performance over adverse transmission channels, considerable efforts have been made ranging from front-end processing, source coding/decoding, channel coding/decoding, packetization to error concealment (EC) [41]. A diagram of a typical DSR system is shown in Fig. 3.3 below.

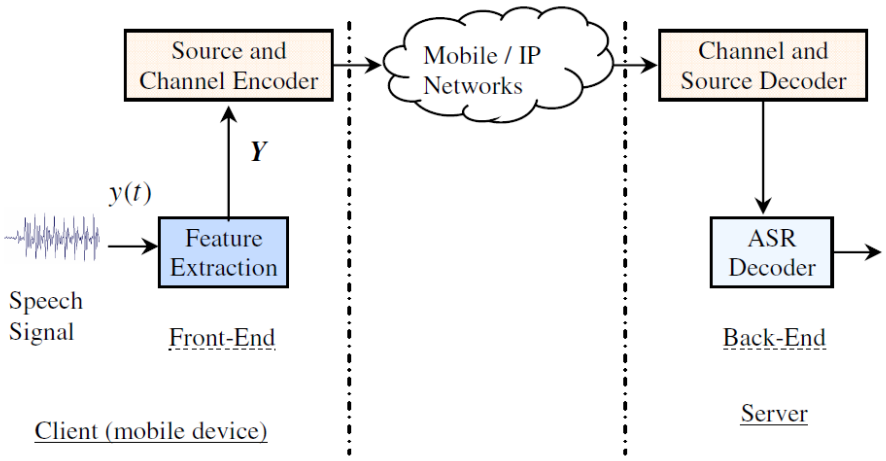


Figure 2.6 Architecture of a distributed speech recognition system [38].

Research in DSR has been focused on source coding, channel coding and error concealment. A very promising source coding technique is a histogram-based quantization (HQ). It is motivated by the fact that acoustic noise may move feature vectors to a different quantization cell in a fixed VQ codebook and thus introduce extra distortion. The histogram-based quantization dynamically defines the partition cells based on the histogram of a segment of the most recent past values of the parameter to be quantized. The dynamic quantization method is based on signal local statistic, not on any distance measure, nor related to any pre-trained codebook. The method has shown to be efficient in solving a number of problems associated with DSR including environmental noise, coding, noise robustness and transmission errors [42].

2.8.1.3 Embedded Speech Recognition

Commonly, embedded speech recognition refers to a technique in which all speech recognition processing is located in the target mobile or handheld consumer device. That is the case if no

network connection is available and also for certain speech recognition applications even when a communication link is available, while others may use NSR and DSR methods [3].

From a system architecture point of view, embedded speech recognition may be considered as the simplest approach when implementing speech recognition. In contrast to network or distributed speech recognition, there is no signal or data sent from the client device to a remote server based engine. Hence the application is always ready to use, irrespective of radio link existence and conditions. Given that, it becomes immediately clear that there is a price to be paid for the architecture simplicity: the complex speech recognition algorithm has to run on a generically low-resourced consumer device. Also, all maintenance and upgrading activity falls on the user or service of the consumer device [3].

Fortunately, continuous advance in semiconductor technology implies a rapid evolution of computing speed of microprocessors and improvement of power consumption of memory devices. So, the complexity of speech recognition algorithms is expected to become less and less of a bottleneck in the future when implemented in an embedded manner. Nevertheless, server-based speech recognition will always have an advantage in terms of available resources [38].

2.8.1.4 Comparison of the three Architectures

From the above discriptin of the the three arctitectures, one can infer that, each of the architectures have their own advantates and disadvantages. Tan and Lindberg [38], have summarized the pros and cons of each of the architectures as shown in Table 2.4 below. Application scenarios are listed in the table, though the boundary between them is indistinct, and the question of which type to deploy will highly depend on the application and resources available.

	NSR	DSR	ESR
Network dependence	Yes	Yes	No
Transmission impairment	Yes	Yes	No
Coding effect	Yes	No	No
Computation complexity on device	Very low	Low	High
Memory footprint on device	Very low	Low	High

Battery consumption	Very low	Low	High
Portability	High	Medium	Low
Recognition performance	Compromised	Maintained	Compromised
Application scenarios	Telephone and VoIP applications, SMS dictation, voice search	Application scenarios	Telephone and VoIP applications, SMS dictation, voice search

Table 2.6 Comparison of the three architectures [38].

2.8.2 Speech Recognition Challenges in the Mobile Context

The challenges in deploying automatic speech recognition onto a mobile device are significant. The most notable challenge is the limited computational resources available on mobile devices. In comparison with contemporary desktop computers, mobile devices are inherently featured with compromised computing power, reduced CPU (central processing unit) clock, limited-speed memory access, small memory size and limited battery life. And the hope lies in the continuous advance in semiconductor technology implying a rapid evolution of computing speed and memory size so the complexity of ASR is expected to become less and less of a bottleneck in the future[42].

When it comes to real-world deployment of ASR technology on mobile phones, algorithmic attention also needs to be given to the battery life and energy aware speech recognition as was done in [42]. Speech recognition by machine is not perfect and errors inevitably appear in the recognition results. However, error correction is a nontrivial problem, especially for mobile devices with a small screen. This highlights the importance of user interface design for both voice and GUI and of error correction through multiple modalities, which was shown to be more efficient in [43].

Another challenge related to using ASR systems on mobile phones is the issue of speech data collection. Most speech databases available are collected using either telephones or high-quality headphones whose characteristics are considerably different from that of integrated microphones on mobile devices. Also, microphones on mobile devices are used in a different way and close-talking microphone setups are not convenient for the user. There is a significant lack of databases

collected using mobile devices and the mismatch in training and test (application) acoustic data is known to degrade the speech recognition performance [38].

Mobile devices are often used in varying noisy environment and while on the move. Robustness against noise and degradation in speech quality is difficult to achieve under such conditions and must also be attained with limited resources of mobile devices.

2.9 Related Work

The first research work in the area of automatic speech recognition for Amharic was Solomon's [7] isolated Consonant-Vowel syllable recognition system done in 2001. Thereafter, many researches have been done by different researchers and with different perspectives. The only attempt to incorporate a speech recognition technology with a mobile phone in an offline mode is Tinbit's [8] work. Tinbit has demonstrated the usability of Automatic Amharic Speech Recognition system to command mobile devices shipped with an Android platform. Hence, the researcher developed an android mobile application that can listen to predefined Amharic commands and perform actions accordingly like making a call, opening an email and opening Google. For this, she used small vocabulary consisting of 19 Amharic words. From which, 10 of them are used to recognize the Amharic digit words from 0 to 10 and the remaining words are used to form command words in a connected manner i.e. "Google kefete" and "Tiri astelalf".

Tinbit [8] has adapted the English acoustic model for Amharic language and also used a Cross-language adaptation technique to map or represent the Amharic language sounds/phonemes using the English language phone set. The Speech recognition system developed is a small vocabulary read speech recognizer which is done based on the popular Hidden Markov Model (HMM) concept. The HMM core parts (acoustic model, lexical model and language model) were trained using recorded speech corpus and their transcriptions. To implement the HMM concept, Pocketsphinx library which is under the CMUSphinx Toolkit family and which is a light weight speech recognition engine specifically designed for mobile and handheld devices is used. The accuracy of the Amharic speech recognizer application is tested by five voluntary people and each were requested to say each of the commands and also randomly say three distinct phone numbers to test the dialing functionality. After testing, accuracy of the android mobile speech recognizer was measured to be 72%.

Another work that relates to Amharic speech recognition system and mobile phone is Mikhael's [9] work of 2015. As opposed to Tinbit [8] which is an offline, all the recognition tasks are carried out on a personal computer and then the recognized data will be transferred to an Android mobile phone through a GSM, which then be used for the phone dialing activity.

On his work, Mikhael has designed a small vocabulary, speaker dependent and continuous Amharic ASR that can recognize digit strings and a limited set of names of individuals and organizations spoken continuously to a personal computer. For this, two additional components namely dialog manager and phone dialer are incorporated. Once the speech recognizer correctly recognizes the key words, the output of the recognizer will be integrated to phone dialing application using the dialog manager. The dialog manager receives the recognized textual representation and look up the associate phone number and invokes the phone dialer to make a call.

Mikhael [9] has mentioned that, the Amharic speech recognizer were designed based on phoneme, triphone and tied state triphone acoustic models and two different language models namely, parsed task grammar and backed-off bigram language models are designed in order to test their correlation with performance of the recognizer. Mikhael [9] experiments yielded an accuracy of 99.38% word correctness, 98.75% word accuracy and 90.0% sentence correctness by triphones model.

Although, there are a number of research works on the area of speech recognition for Amharic language, an intensive investigation of literatures reveals that, the two research works mentioned above are the only works that relate to incorporating a speech recognition functionality with a mobile phone using Amharic language. However, one research work that relates to applying an Amharic ASR system with a computer application is Martha [44] study of 2003, which is Application of Amharic Speech Recognition System to Command and Control Computer: An Experiment with Microsoft Word. Hence, Martha's [44] study is presented below and following, some other research works done for other languages in the area of ASR system for mobile phones are presented.

Martha [44] explored the possibility of developing an Amharic speech input interface to command and control Microsoft Word. Martha has developed a HMM variable variance based, speaker independent, small vocabulary, isolated Amharic word recognizer and used for the development

of the prototype Amharic speech input interface system. The variable variance based models used because they showed better recognition performance than fixed variance based models.

In Martha [44] work, 50 command words were selected from different menus (File, View, Insert, Tools, Table, Window, and Help), translated to Amharic. And speech data of these 50 command words were recorded from 26 people (10 female and 16 male) in the age range of 20 to 35. From the recorded data, 76.9% were used to train the recognizers and the remaining for testing the performance of recognizers.

The communication interface has been written using Visual Basic 6. To test the performance of the system as a whole, 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. Generally, Martha's [44] study of 2003, has demonstrated the usability of whole word modeling for developing Amharic speech recognizer that can be used as a component of speech input interface.

Hugeng1 and Hansel [45], investigated the use of Indonesian speech recognition system to develop Indonesian geography dictionary application on a mobile phone with an Android operating system. The application uses a smartphone to receive input in the form of a spoken word from a user. The approach H. Hugeng1 and E. Hansel used for developing the recognizer is Hidden Markov Model which is contained in the Pocketsphinx library. The application can be used without internet access or in an offline mode, which is pointed out as a one of the good characteristic of the application.

In data collection process, the researchers listed the words which are list of geographic terms; and each word is then chopped into phonemes. These phonemes are included in a phonetic list. The phonemes used in H. Hugeng1 and E. Hansel work are English phonemes which are adopted as Indonesian phonemes. A list of sentences containing all the selected words is made in a text file and each sentence line consists of ten words that could be repeated. The list of sentences is processed using Sphinx Knowledge Base Tool to produce language model for training.

The researchers carried out the testing activity with four conditions to determine the level of accuracy. The four conditions are near silent, near noisy, far silent, and far noisy. From the testing

and analysis conducted, it can be concluded that the developed application can be built as a speech recognition application on Android for Indonesian geography dictionary with an average word recognition accuracy of: 52.87% for near silent condition, 14.5% for near noisy, 23.2% for far silent and 2.8% in the far noisy condition.

Mulhern et al. [46], investigated the use of a speech recognition system and develop assistive technology to help persons with disabilities. In their project, the researchers explored voice recognition as a template upon which the independence of persons with neuromuscular disorders can be expanded. For this, two Android applications were developed on a smart phone to operate a television remote via an input/output (IOIO) board. The developed voice-controlled applications are online and offline, the reason for this is to insure the accessibility of the applications by those with and without Wi-Fi and/or cell phone data plans. For the online or the internet dependent application the researchers used Google's Voice Recognition system and for the independent application (offline) CMU's Pocketsphinx is used.

Both the online and offline applications were tested for response accuracy with respect to distance from the user's mouth and surrounding noise. Each Android application functioned to recognize specific keywords, send appropriate signals to the IOIO board then trigger the remote. There were seven keywords (commands) in total: channel increase, channel down, volume increase, volume down, power on, and off. The "power" command for example, controls the cable box while "on" and "off" control the television. Each of the seven commands are tested in different settings for accurate recognition using both applications.

Though the researchers did not show the recognition performance of their system in numbers, they have clearly illustrated the results with graphs. The graphs show, a general decrease in the proportions of correct recognition when the distance of the phone to the subject is increased as well as when level of ambient noise is increased, which is true for both the online and offline applications. In general, the test results show that both online and offline applications provide quality recognition and execution of desired commands within a reasonable distance and level of surrounding noise.

Liuxinfei and Zhouhui [47], investigated the use of a small vocabulary offline Chinese speech recognition system that can recognize a set of common voice commands used in voice controlling

tasks in Android smartphone. Some of the Chinese voice commands used as a voice controlling activity in the study include Play music, Turn off the music, Open the browser, Exit the browser, Make a phone call, Send SMS and the likes.

For the acoustic model, the researchers adapted the existing English acoustic model using a one person voice in order to improve recognition rate of the system and to fit for user's voice. The corpus contains several sentences which can be arbitrarily assigned but cover words or phonemes to be recognized.

The testing results show that the system based on Pocketsphinx developed in the Android development environment has excellent recognition reaching a recognition accuracy of 90% WER for quiet laboratory environment. The researchers also have revealed that the recognition rate is extremely affected by user's accent because only one person's voice data is used for adaptive training of acoustic model and the system is easily to make mistakes because of high sensitivity to noise around.

Chapter Three

System Design and Architecture

3.1 Introduction

In this chapter, the approach, tools, techniques and procedures used in this research are discussed in detail. As mentioned earlier, the methodology chosen to implement the speech recognizer that converts the spoken Amharic words to a text on a mobile phone is the most popular statistical model, which is the HMM. A detailed discussion of HMM is presented under Section 2.6.

3.2 Corpus Preparation

Speech corpus is one of the fundamental requirements of any speech recognition system. Thus, as mentioned earlier, a speech corpus of all the selected words is recorded from one person using PC and also a mobile phone using the appropriate software. The reason for recording the audio data using both the PC and mobile phone is, to have a representative speech. With this respect, Tan and Lindberg [3] stated that, an audio recorded with high-quality headphones has considerably different characteristics from that of an audio recorded using the integrated microphones on a mobile phone and also, microphones on mobile devices are used in a different way from that of the microphones on computers. Hence, in this research we recorded the audio data using PC with a goal of getting a quality audio, and using a mobile phone with a goal of capturing the environment and context that the ASR system is going to be used, which is on mobile phones.

CMUSphinx website [15] states that, to train a speaker dependent model and a model that can be used for command and control functionalities, a one hour audio data is enough. Accordingly, a total of 1,218 sentences, containing multiples of words in different combinations are created and a 1 hour and 18 minutes audio data of these sentences is recorded. A Sample of the sentences, is presented under Appendix 3.

The recording is carried out in a semi-noisy environment. The recording on PC is done using Audacity software and the microphone on Sony MDR-663MV headset. On the mobile phone, the recording is done using Easy Voice Recorder application and the built-in microphone on a Huawei G730-U00 mobile phone. The settings used while recording the audio on both the PC and the mobile phone are shown in table 3.1 below:

Format	Sample rate	Depth	Chanel
MS WAV	16 kHz	16 bit	mono

Table 3.1 Settings used in recording the audio

After all the sentences recorded continuously, Audacity software is used to segment and to label the audio of each sentence into the corresponding sentence file id. And since, the recorded data is used for training the acoustic models and also for testing their performance, around 82% of the data is used as a training set, and the remaining 18% is used as a test set.

3.3 The Architecture of the System

A speech recognition system can be implemented in many different areas and methods. When it comes to integrating ASR systems with mobile phones, there are three commonly known speech recognition approaches namely, network speech recognition, distributed speech recognition and embedded speech recognition. From these, the third one, which is embedded speech recognition or simply ESR, is chosen in this particular study. In ESR, all speech recognition processing is located in the target mobile or handheld consumer device. In other words, it is fully an offline and can be used at any time and place without the need to have a network connection.

There are several reasons for choosing the ESR architecture in this study. Some of them are,

- It's simplicity of implementation, as there will be no signal or data to be send from the client device to a remote server based engine and vice versa.
- The steady increase in processing power and storage capacity of the mobile phones, enabling all the decoding and recognition activities to be carried out on the mobile phone.
- The currently poor telecommunication services in local context, which can make ASR systems deployed using the NSR and DSR architectures challenging, makes the ESR architecture preferable.

Hence, the architectural design of the ASR system designed in this study resembles much the architecture of an ASR system on a PC as it incorporates the same components of the general

speech recognition system. Fig. 3.1 depicts the diagrammatical representation of the developed Amharic ASR system for mobile phones.

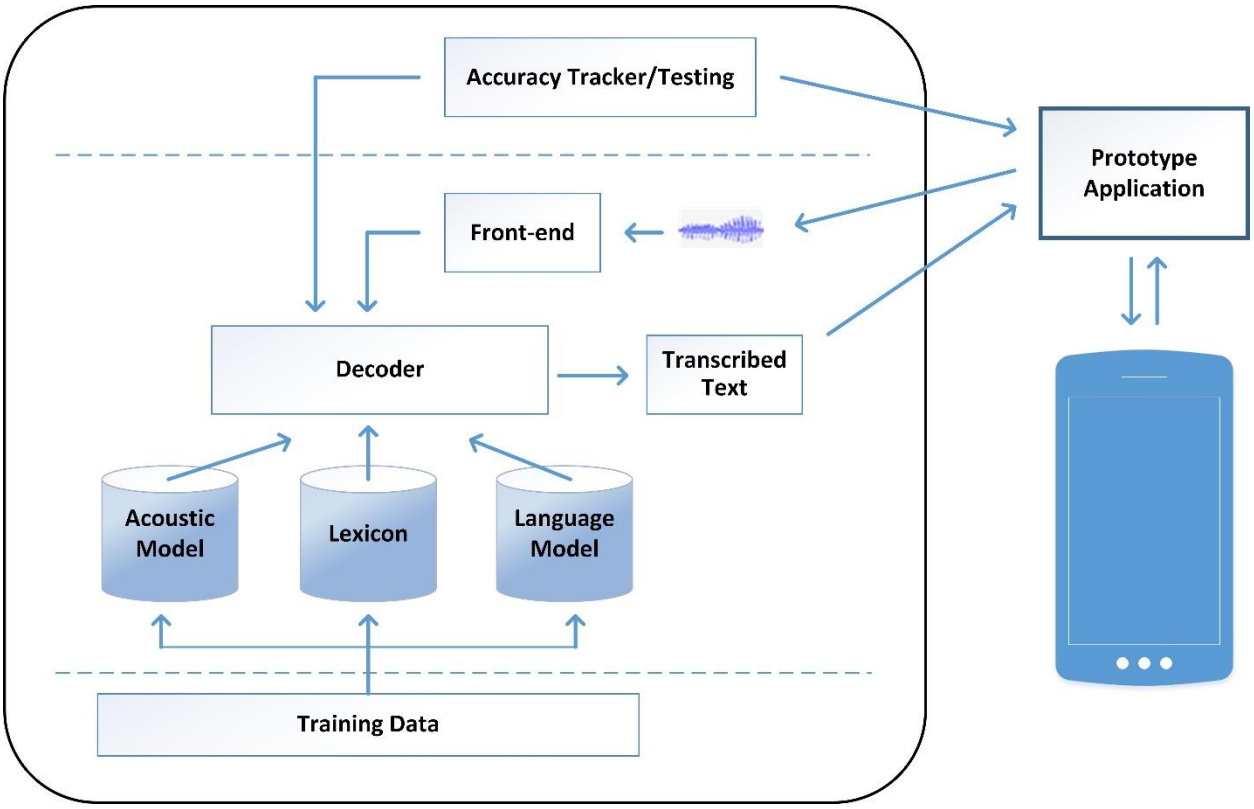


Figure 3.1 The Amharic ASR system designed for mobile phones

3.4 Components of the ASR System

3.4.1 Front-end

The front-end component involves signal processing and feature extraction. The feature extraction of the ASR system maps the speech waveform into a sequence of feature vectors. This sequence of feature vectors is subsequently used to train acoustic model and decode input speech waveform. In order to find some statistically relevant information from incoming data, it is important to have mechanisms for reducing the information of each segment in the audio signal into a relatively small number of parameters, or features. These features should describe each segment in such a characteristic way that other similar segments can be grouped together by comparing their features. There are many feature representations in use, but the most common is the mel-frequency cepstral coefficient (MFCC) feature set, and in the developed system MFCC is used to extract features of audio used during training the models and also during recognition [23].

The MFCC is the most evident cepstral analysis based feature extraction technique for speech and speaker recognition tasks. It is popularly used because it approximates the human system response more closely than any other system as the frequency bands are positioned logarithmically [48]. Computing MFCC is based on the short-term analysis, and thus from each frame a MFCC feature vector is computed. In order to extract the coefficients, the speech sample is taken as the input and it is divided into number of frames. After that, the hamming window is applied to minimize the discontinuities between the frames where Discrete Fourier Transform (DFT) is used to generate the Mel filter bank. According to Mel frequency wrapping, the width of the triangular filters varies and so the log total energy in a critical band around the center frequency is included [49]. Figure 3.2 below shows the steps involved in MFCC feature extraction.

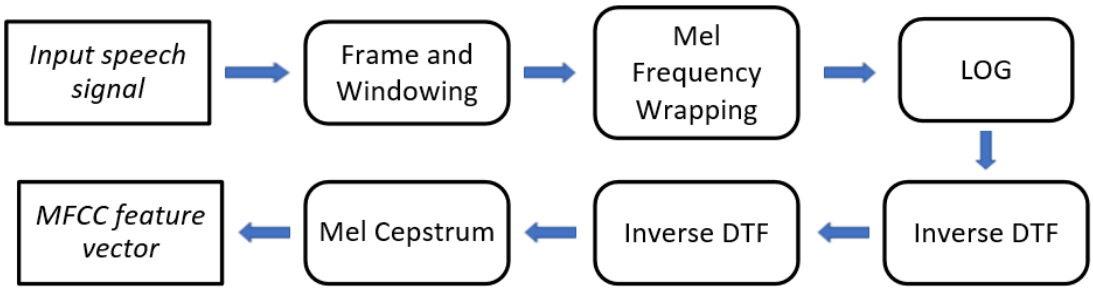


Figure 3.2 Steps involved in MFCC Feature extraction [49]

3.4.2 The Lexicon

In Speech recognition systems lexicon model also called pronunciation dictionary, is mapping of all the words to their corresponding pronunciation forms in terms of the recognition unit (i.e. phone, sub-phone, or syllable) sequences. It specifies the finite set of words that may be produced by the speech recognizer and gives, at least, one pronunciation for each.

In this research work, to train the two acoustic models, word-dependent phone (WDP) and word-dependent CV syllable dictionary (WDCVS), 36 distinct words are employed. This number of words is found to be enough for the research, as it investigates the use of small vocabulary ASR system that can be used to command and control mobile phones. And to select these words, the researcher used his own observation.

From the 36 words, 15 of them which in combination can make 11 commands as they are spoken continuously i.e. “ሙዚቃ ክፈት”, “ሙዚቃ አጫውት” or “ኦፔራ ክፈት”, which means open music, play

music and open Opera, are used to train the system to recognize Amharic command and control phrases. The remaining 21 words which in combination can represent Amharic digits from "ዜሮ" up to "መቶ", which means form zero up to hundred, making a total of 101 digit words as they are spoken continuously, are used to train the models to recognize the 101 Amharic digits. For instance, words "አንድ", "ሁለት", "ሶስት" which means one, two, and three, can be used to form the numbers "አስራ አንድ", "አስራ ሁለት", "አስራ ሶስት" which means eleven, twelve and thirteen respectively. Likewise, the same digits can be combined with "ሀያ" to form other digit words, i.e "ሀያ አንድ", "ሀያ ሁለት", "ሀያ ሶስት" which means twenty one, twenty two, twenty three respectively.

In both the pronunciation dictionaries, WDP and WDCVS, all the 36 distinct words are transcribed, and intra-speaker variabilities are not considered; in other words, only one transcription is used for each word in the lexicon. Also, the two dictionaries do not handle the variation of the pronunciation of the sixth order grapheme and the difference between geminated and non-geminated consonants.

3.4.3 The Acoustic Model

The Acoustic model refers to the statistical representations for the feature vector sequences computed from the speech waveform. HMM, among others, is the most common type of acoustic model, and in this research work HMM is used for acoustic modeling. Acoustic modeling also encompasses "pronunciation modeling" which describes how a sequence or multi-sequences of fundamental speech units (such as phones or phonetic feature) are used to represent larger speech units such as words or phrases that are the object of speech recognition [18].

According to the CMUSphinx website [15], CMUSphinx system does not support whole-word based models rather, word-dependent phone (WDP) models and the developers claim that these models are equivalent with whole word-based models or even better sometimes. It is indicated in the website that, word-dependent phone model is created by creating word-dependent phone dictionary, and an example illustrates word-dependent phone dictionary is created by attaching the word in the dictionary with each phone in the word with an underscore and then using them as the sequence of phones for that word in the pronunciation dictionary. Accordingly, the word-dependent dictionary is created. To get the phonemes for creating this dictionary, the transliteration scheme proposed by [50], that is, using ASCII characters for the language graphemes, is employed.

Sebsbie et al. [50] stated that, in Amharic language the CV syllables, among others, cover the large majority of syllable distribution. And, [28] stated that, the use of CV syllables is a promising alternative in the development of ASR systems for Amharic. In addition to this, [51] stated that, CV syllable based recognizers outperformed the tied state Triphone based recognizer in accuracy, and therefore, the use of CV Syllables is a promising direction for Amharic speech recognition.

Accordingly, the Amharic CV syllables are used to create another dictionary, word-dependent CV syllable dictionary, by replacing the phonemes in word-dependent phone dictionary with this CV syllables. And thus, in this research work, two pronunciation dictionaries, WDP and WDCVS dictionaries are created (as shown in Appendix 1), and are used to develop the two acoustic models, WDP and WDCVS.

In both dictionaries, the phones which are created by attaching each phone/CV syllable in the word with the word itself by an underscore, are used to form the phone set in the respective models, resulting 194 unique phones for the WDP model and 115 unique phones for the WDCVS model. The Phoneset used in developing WDCVS model is presented in Appendix 4.

To get the best of each of the two models, each model is trained with the three mode types that CMUSphinx supports (continuous, semi-continuous and PTM), and then the recognition performance of each model is computed and compared with each other to decide which one performs best.

3.4.4 Language Modeling

A language model refers to a system's knowledge of what constitutes a possible word, what words are likely to co-occur, and in what sequence. The semantics and functions related to an operation a user may wish to perform may also be handled by a language model. In continuous speech recognition, the incorporation of a language model is crucial to reduce the search space of sequence of words [18]. A language model can either be a statistical language model (SLM), such as an n-gram model, or a grammar based language model, for example a context-free grammar (CFG) or a finite-state automaton (FSA). Statistical language models try to predict all the valid utterances in a language, by combining all the recognized words into every possible combination. Context-Free Grammars are restricted forms of a language model, that restrict the recognized phrases to a predetermined set, and discard those that do not fit that model [52]. While statistical language

models are useful for open-ended applications, like dictation and general-purpose recognition, context-free grammars are suitable for specific applications, like command-and-control systems.

Thus, in this research work to model the sequence of acceptable words in two different contexts i.e. in either phone dialing or commanding, three separate context-free grammars are created. The first grammar is for the digit words, the second is for the command phrases and the third which served as a menu, is used to switch between the two grammar files – the digits and commands. All the three grammars are created based on the Java Speech Grammar Format (JSGF). According to [53], JSGF is a platform-independent, vendor-independent textual representation of grammars for use in speech recognition to determine what the recognizer should listen for, and so describe the utterances a user may say.

The digits grammar is used to describe the acceptable single or connected word from "ዜሮ" to "ሙዶ". In these set of digit words, some of the digits are independent/single words and others are formed in a connected manner which correspond to the larger proportion, i.e. numbers beyond 10 with the exception of multiples of 10. Any digit from the single or the connected digits set can be uttered at any time, and this is handled by the digits grammar in such a way that, if a valid single word digit is spoken, that respective word will be searched and the search ends. However, if the spoken digit is a connected word digit, the first word will be searched first and then the second word will be searched and matched with the first one and the search ends.

Fig. 3.3 below shows the JSGF grammar used to model the Amharic digits words from "ዜሮ" to "ሙዶ" and Fig. 3.4 illustrates the possible paths graphically.

digits.gram | commands.gram

```
#JSGF V1.0;

grammar digits;

<i1> = ዜሮ | አንድ | ሁለት | ሶስት | አራት | አምስት | ስድስት | ሰባት | ስምንት | ዘጠኝ | አስር |
       ሀያ | ሰላሳ | አርባ | ሀምሳ | ስልሳ | ሰባ | ሰማንያ | ዘጠና | መቶ | ደውል | አቋርጥ | አጥፋ ;

<i2> = አንድ | ሁለት | ሶስት | አራት | አምስት | ስድስት | ሰባት | ስምንት | ዘጠኝ ;

<i3> = (አስራ)(<i2>) ;

<i4> = ሀያ | ሰላሳ | አርባ | ሀምሳ | ስልሳ | ሰባ | ሰማንያ | ዘጠና ;

public <digits> = [ <i1> ] | [ <i3> ] | [ <i4> ] [ <i2> ] ;
```

Figure 3.3 JSGF grammar used to model the Amharic digits from "ዜሮ" up to "ሙዶ".

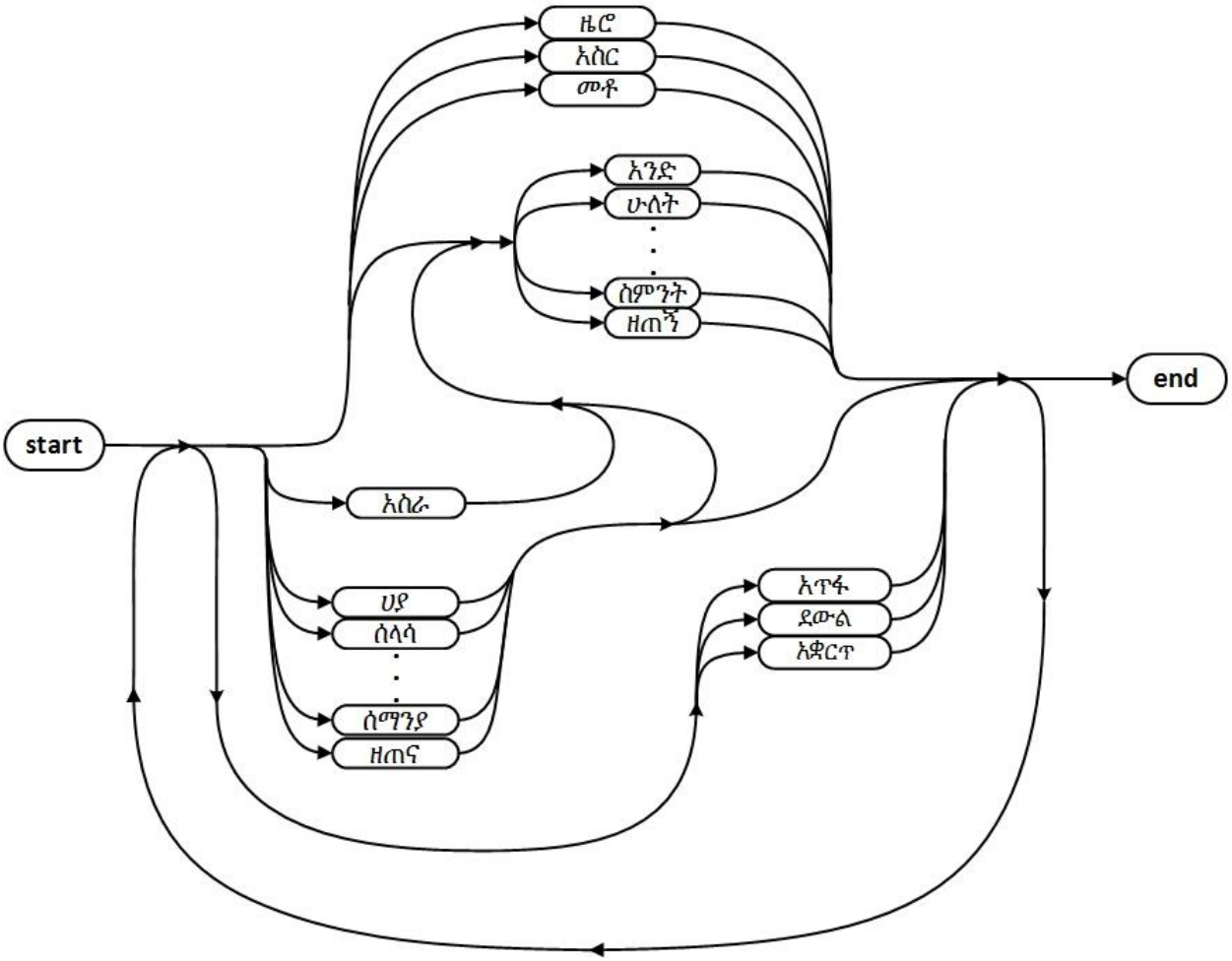


Figure 3.4 Graphical illustration of the possible paths in forming Amharic digits from "ዜሮ" to "መቶ"

There are two reasons why this method is employed instead of simply including all the 101 digit words in the JSGF grammar file. The first is, it limits the number of alternatives in the grammar which in turn reduces the searches that the system needs to undergo. And the second is, this grammar can be easily extended to incorporate the other numbers beyond 100. For instance, the Amharic digit words from 101 to 999 can be simply added to the grammar without increasing the lexicon, just by altering the grammar to handle the word "መቶ" with its acceptable right and left digits which are already in the grammar.

For the command and control words, another JSGF grammar file is created. As it is shown in Fig. 3.5 below, the grammar file contains 10 distinct words and can be used to model 9 command

phrases that are common in mobile phones. Fig. 3.6 graphically illustrates the paths in forming a valid command phrase.

digits.gram commands.gram

```
1  #JSGF V1.0;
2
3  grammar commands;
4
5  public <command> = ሙዚቃ ( ክፈት | አጫውት | አቁም | ዝጋ ) | ኦፔራ ( ክፈት | ዝጋ ) | ባትሪ ( አብራ | አጥፋ ) ;
```

Figure 3.5 JSGF Grammar for the Amharic command words used in the recognizer

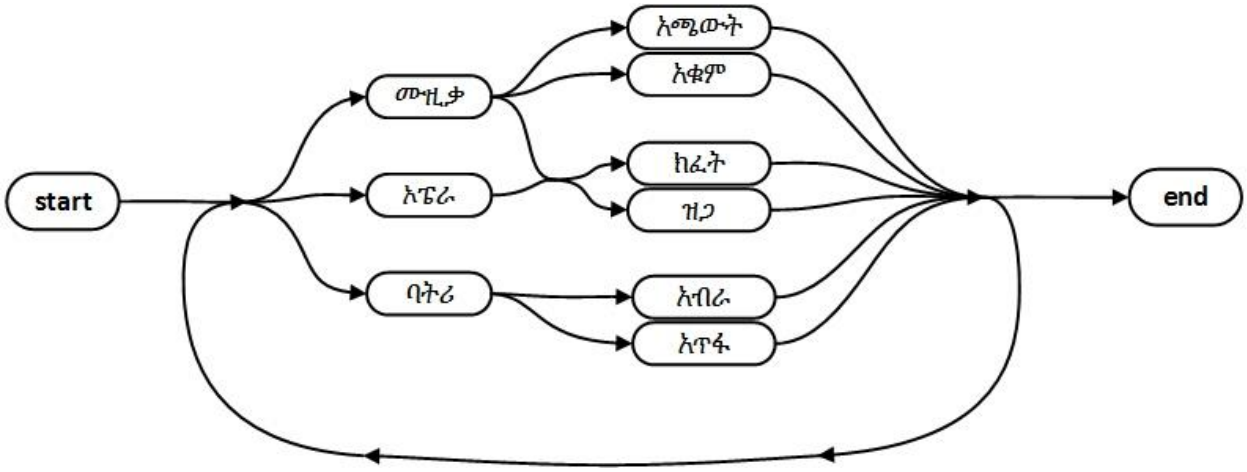


Figure 3.6 Graphical illustration of the possible paths to form a valid command phrase

Since it is possible to switch between searches at the time of recognition in the Pocketsphinx, a third JSGF grammar is also created and employed in the prototype application to switch between the two grammars files – the digits and the commands. Hence, this grammar has only two alternatives, the two phrases that the user need to say i.e. "በድምፅ መደወል" or "በድምፅ ማዘዝ", to go to ether the phone dialing or the commanding functionality.

3.5 Development Tools

3.5.1 CMU-Sphinx toolkit

CMU-Sphinx is a famous speech recognition toolkit used for various speech recognition research works and also for developing several speech based applications. The toolkit is developed at Carnegie Mellon University and includes the speech recognizers – Pocketsphinx and Sphinx-4,

acoustic model trainer – Sphinxtrain and a library – Sphinxbase. According to CMUSphinx [15] website, from the different packages, which one to use depends on the nature of the application. Accordingly,

- Pocketsphinx – lightweight recognizer library written in C.
- Sphinxbase – support library required by Pocketsphinx
- Sphinxtrain – acoustic model training tool

For each of the above packages there are latest versions (5prealpha). And as the website [15] recommends to use these latest releases, in this study, Pocketsphinx-5prealpha is used as a decoder, sphinxtrain-5prealpha is used to train the models and sphinxbase-5prealpha a library needed by Pocketsphinx is used.

The decoder used in this research work which is Pocketsphinx, is an open source speech recognition toolkit optimized to run on embedded devices. Pocketsphinx runs locally on the device; it is fast and requires comparatively modest computational resources. It has voice activity detection functionality so that continuous recognition can be performed. Pocketsphinx can be fully customized for different requirements. It also has the option of customizing the default American English acoustic model by adapting the model using the toolkit SphinxTrain. Pocketsphinx is open-source, comes with BSD license and can be used in commercial applications [15] [14].

3.5.2 Pocketsphinx Android Demo Application

It is a demo application from GitHub, Inc. [54], and it demonstrates how Pocketsphinx is used on an Android mobile phones. The application has sample English acoustic model, Language model, JSGF grammars and also a library compiled using Pocketsphinx. Hence, we have downloaded and installed this demo application on Android Studio virtual emulator phone and experimented with in order to become familiar with Pocketsphinx on mobile phones prior to developing the prototype application. Moreover, we've used some of the source code in this demo application during the development of the prototype application.

3.5.3 Pocketsphinx-android

Pocketsphinx-android is a wrapper for Pocketsphinx for Android providing high-level interface for recognizing the microphone input [55], and in this study, Pocketsphinx-android is used to compile the Pocketsphinx library used in the prototype application. To compile the library, the instructions in the CMUSphinx [15] website are followed.

3.5.4 Android and Android SDK

As mentioned previously, the mobile phones used for integrating the Amharic ASR are, mobile phones that use Android as their operating System. From the several reasons for choosing this platform, the main ones are, its dominance both in the market and the community, its adaptability and also its capability to support a myriad of applications developed by developers all over the world including an ASR system developed using the CMUSphinx speech recognition toolkit.

Android is a Linux-based operating system designed primarily for touch screen mobile communication devices such as smartphones and tablets, though variants of Android are also exist on different devices i.e. game consoles, televisions, wrist watches. Android was developed by Google in collaboration with Open Handset Alliance and supports open source projects so that many Android developers can build their own Android-based applications using the tools Android Software Development Kit (SDK) and Native Development Kit (NDK) which can be obtained free of charge [46].

The Android SDK includes a comprehensive set of development tools. These include libraries, a handset emulator, documentation, sample code, tutorials & tools such as dx - Dalvik Cross - Assembler, Android Asset Packaging Tool and Android Debug Bridge. Applications are written using the Java programming language and run on Dalvik, a custom virtual machine designed for embedded use which runs on top of a Linux kernel. The mobile device emulator lets the developer prototype, develop, and test Android applications without using a physical device. The Android emulator mimics all of the hardware and software features of a typical mobile device, except that it cannot receive or place actual phone calls [56].

Hence, the latest version of Android SDK, version 2.3.3, is used to develop the prototype application that served for integrating the Amharic ASR system with mobile phones. The

application is designed and built using Java programming language in accordance with the programming language guidelines for developing applications on Android operating system (Appendix 5). After the prototype application is created, the necessary libraries, the developed acoustic models and all the necessary files i.e. dictionary and the grammar files, are imported to the application and the prototype Amharic ASR system for mobile phones is created. Following, Android Studio's virtual emulator phone, Nexus 4 with API level 22, is used to install the prototype application and also to run and test its functionality, as shown in Fig. 3.7, before installing and conducting the performance evaluation on a real/physical mobile phone.

3.6 The Prototype Application

The prototype Amharic ASR application for mobile phones primarily developed to demonstrate the usability of the models and also to conduct an empirical performance tests on the models. In view of this, the prototype application starts by first asking whether dialing or commanding by voice is need to be conducted and waiting for an audio input signal. Based on the recognized word the application automatically switches to either the digits or commands grammar files using the third grammar file and again starts to listen for a spoken Amharic digit word or command phrase. Finally, using the respective grammar file i.e. digits or commands, the prototype application recognizes the spoken word or phrase, and then either displays it using a TextView[1] or a Toast[2], or performs the appropriate task.

Here, the act of asking for an input, accepting the input and then performing the appropriate activity based on the accepted input data can be likened to any other command and control activity where the user commands and/or controls his mobile phone using his voice. And thus, this achieves the objective of this research, that is, investigating ways of integrating an Amharic speech recognition system that can be used to command and control mobile phones.

After testing the operability of the prototype Amharic ASR application on a virtual emulator using all the developed acoustic models, a physical mobile phone is connected to Android Studio and

[1] A user interface element that displays text to the user in Android.
[2] Used to notify user about an operation with a small popup that displays for a small period of time, without expecting any user input.

the prototype application is installed on a physical mobile phone with each of the trained models and the performance of each model is evaluated, as presented in detail in Section 4.3.2.

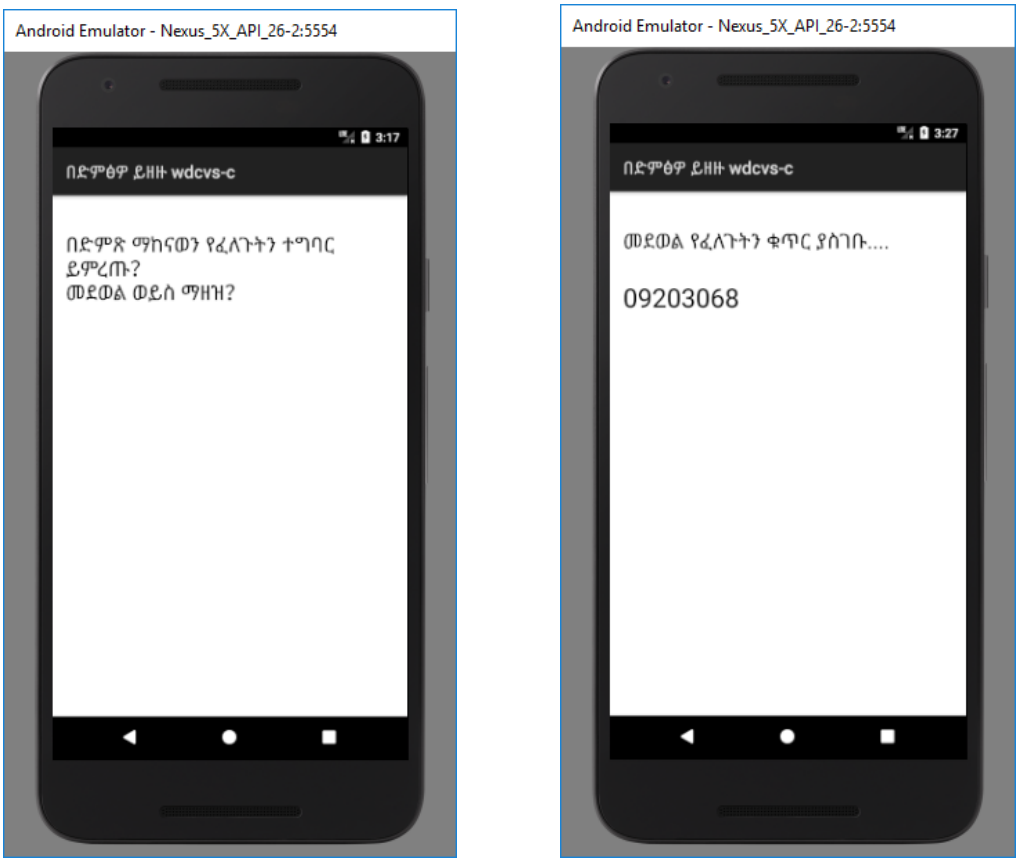


Figure 3.7 Screenshots of the Prototype Amharic ASR system on Android emulator phone, Nexus 5X with API level 26

Chapter Four

Experiment and Discussion

4.1 Introduction

This chapter presents the experiments and the procedures followed while conducting the experiments in this research together with the results and discussion of the results. As the objective of this research is investigating an enhanced way of integrating an Amharic ASR system on a mobile phone, 36 distinct Amharic words are employed to build two acoustic models namely, word dependent phone model and word dependent CV Syllable model. Thus, the speech corpus of these words is recorded from one person and used for both training and testing the models. For this, the corpus is divided into training and testing stacks with the training stack receiving around 82% of the data and the testing stack the remaining 18%. All the training and testing conducted are presented as follows.

4.2 Training the Models

A Hidden Markov Model based system, like all other speech recognition systems, functions by first learning the characteristics or parameters of a set of sound units, and then using what it has learned about the units to find the most probable sequence of sound units for a given speech signal. The process of learning about the sound units is called training and the process of using the knowledge acquired to deduce the most probable sequence of units in a given signal is called decoding, or simply recognition.

CMUSphinx supports three different types of acoustic models: continuous, semi-continuous and phonetically tied (PTM). Hence, the two models used in this research work namely, WDP and WDCVS, are trained using the three model types to decide which of the model type perform best on each of the two models, and therefor to select the best one. Consequently, six different experiments are carried out; three for each of the WDP and WDCVS models.

The difference of the three model types, according to CMUSphinx [15] website is, the number of mixture of gaussians used to compute the score of each frame. In continuous model, every senone has its own set of gaussians, thus, the total number of gaussians in the model is about 150 thousand

which is too much to compute the mixture efficiently. In contrary to the continuous, in semi-continuous model, to compute the score of each frame 700 gaussians are used, which is small and therefore, used only with different mixtures to score the frame. Due to the smaller number of gaussians semi-continuous models are fast, but because of more hardcoded structure they are also a bit less accurate, and this makes PTM models a good alternative. PTM models use about 5000 gaussians, thus providing better accuracy than semi-continuous, keeping the speed of continuous model types.

However, since the effect that these three model types have on the WDP and WDCVS models used in this research work, is not known beforehand given the corpus, and because each of the three model types could be possible alternatives to train acoustic models for mobile phones, each of the three model types are used to train the two models. This is important to investigate the effect the three model types have on the two models and also to select and incorporate only the models with good recognition accuracies into the prototype application if significant recognition accuracies are observed. However, if no major recognition performance difference is observed among the trained models, all the models will be incorporated into the prototype application and testing will continue on mobile phone for all the trained models. It is because, minor recognition accuracy variations may not persist in mobile phones and therefore may not have influence on the recognition accuracy on mobile phones – the devices in which the models are intended for.

However, despite the fact that, the number of tied states (senones) and the number of Gaussian mixture distributions (densities) used in the HMM can have a significant impact on the models' accuracy, these parameters are not considered in this research, and therefore, all the six training activities conducted are carried out with the recommended approximate values, shown in Table 4.1 below. The reason for not considering these parameters and experiment with to get optimal recognition accuracies is that, doing so will obviously increase the number of experiments need to be conducted, which in turn will take significantly much time that can be beyond the time limit of this research work. Moreover, models trained with the approximate recommended values, have achieved excellent recognition accuracies.

The recommended approximate number of senones and the number of densities for a continuous model in different scenarios, according to CMUSphinx [15], are shown in table 4.1 below. For

semi-continuous and PTM models, the website recommends using a fixed number of 256 for densities.

Vocabulary	**Audio in database / hours**	**Senones**	**Densities**	**Example**
20	5	200	8	Tidigits Digits Recognition
100	20	2000	8	RM1 Command and Control
5000	30	4000	16	WSJ1 5k Small Dictation
20000	80	4000	32	WSJ1 20k Big Dictation
60000	200	6000	16	HUB4 Broadcast News
60000	2000	12000	64	Fisher Rich Telephone Transcription

Table 4.1 The approximate number of senones and the number of densities for a continuous model [15]

From the table above, it can be seen that the vocabulary size and the length of the audio data that is somehow closer to the one used in this research is, the first one which is Tidigits Digits Recognition, that uses 200 for senones and 8 for densities. Therefore, in all the six experiments, 200 is used as a number of senones. And for the number of densities, 8 is used for training the continuous models, and 256 for semi-continuous and for the PTM models. Table 4.2 below summarizes the number of senones and densities used to train the two models, WDP and WDCVS, with the three model types.

Model Type	Senones	Densities
continuous	200	8
semi-continuous	200	256
PTM	200	256

Table 4.2 Number of senones and densities used to train the models

As mentioned in the previous chapter, in this research work, three grammar files are created for the two sets of words i.e. digits and command. However, for the sake of convenience, the three grammar files are combined into one at the time of both the training and decoding/testing phases, since doing so does not have impact on the result.

After all the environment variables and directory structure is set, the training is started by running the command *sphinxtrain -t db_name setup*, where, *db_name* is the name of the database. This command creates a configuration file which can be used to setup the different parameters, including the ones discussed above.

After configuring all the necessary settings and values, i.e. the number of senones, densities, the model type and file paths in the configuration file created using the above command, the second command, *sphinxtrain run*, is executed to start the actual training.

At the end of the training, several new directories are created that contain files which are generated in the course of the training. The decision tree built for "a_ʜʌɳʕ" while the training phase of the WDCVS model with continuous model type, for example, is presented in Appendix 6. An html file is also created, with the named *db_name.html*, and contains status report of the executed jobs. This file is used to verify what jobs are launched and completed successfully. The trained models for all model types will be stored in *model_parameters* directory under the database directory.

Figure 4.1 and 4.2 below, show the status reports generated after training the WDP model with continuous model type and WDCVS model with continuous model type, respectively.

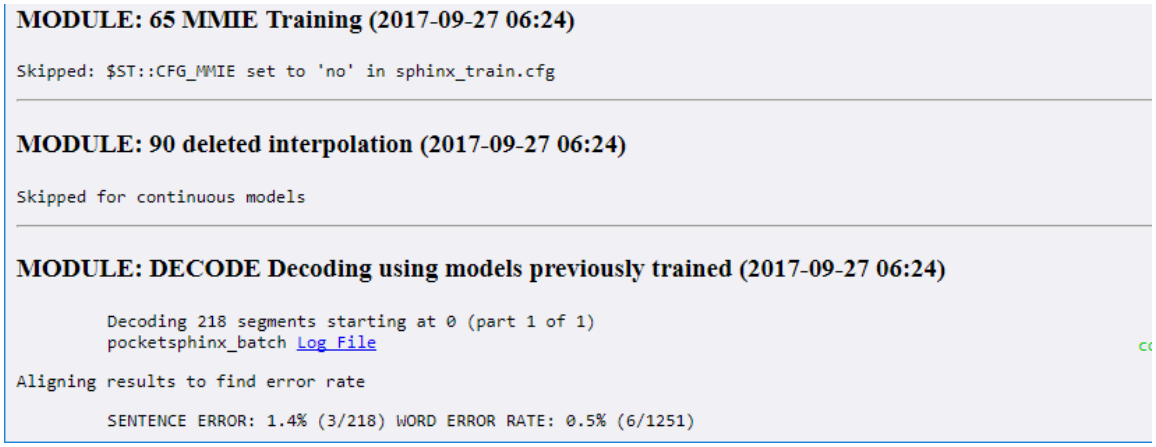


Figure 4.1 Status report generated after training the WDP model with continuous model type

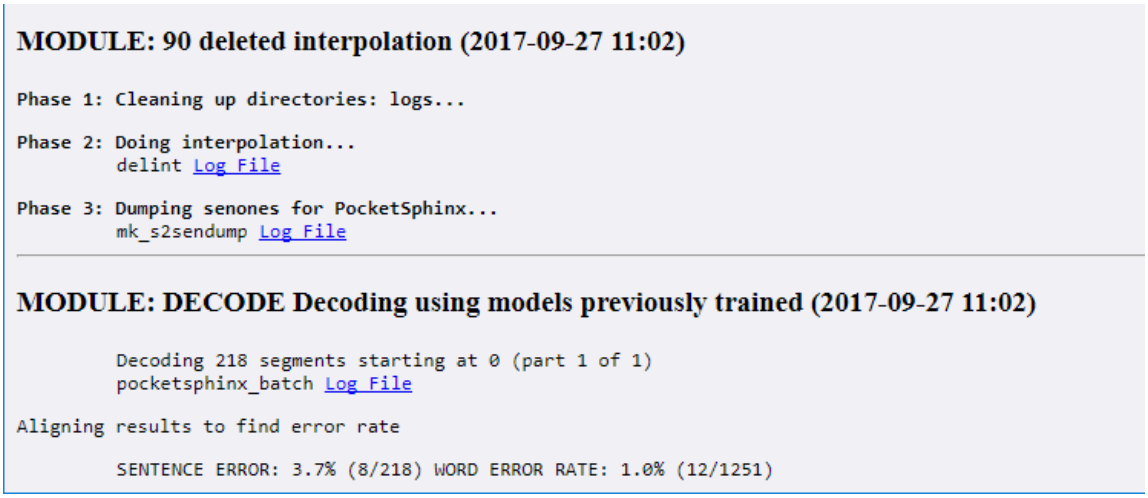


Figure 4.2 Status report generated after the completion of training the WDCVS model with PTM model type

All the training activities (and also the testing activities in section 4.3.1) are performed on a Toshiba Portage 980 laptop computer with 8GB RAM and core i5 CPU model with 2.6GHz speed installed with Ubuntu 16.04 LTS 64 bit. The compiler gcc v5.3.1, the decoder Pocketsphinx-5prealpha (v0.8), a trainer Sphinxtrain-5prealpha (v1.0.8) and a library Sphinxbase-5prealpha (v0.8) are used to train the models; all from the official repositories. The results of the trained models are presented and discussed in the section 4.3 below.

One problem encountered during this training phase is, the number of phones used in developing the WDP models which is 194 (as discussed in section 3.4.3), is greater than that of the maximum number of phones which is 128, specified in the Pocketsphinx source code, and because of this, the decoding at the end of the training and also the decoding carried out in section 4.3.1 for the WDP models couldn't be carried out successfully and the trainer/decoder exits out with a similar error message shown in Fig. 4.3 below. Furthermore, this problem did not allow the WDP models to be run on both the virtual emulator and the physical mobile phones, since the library used in the prototype application is compiled using Pocketsphinx and Pocketsphinx-Android (discussed in Section 3.5.3), with this maximum number of phones.

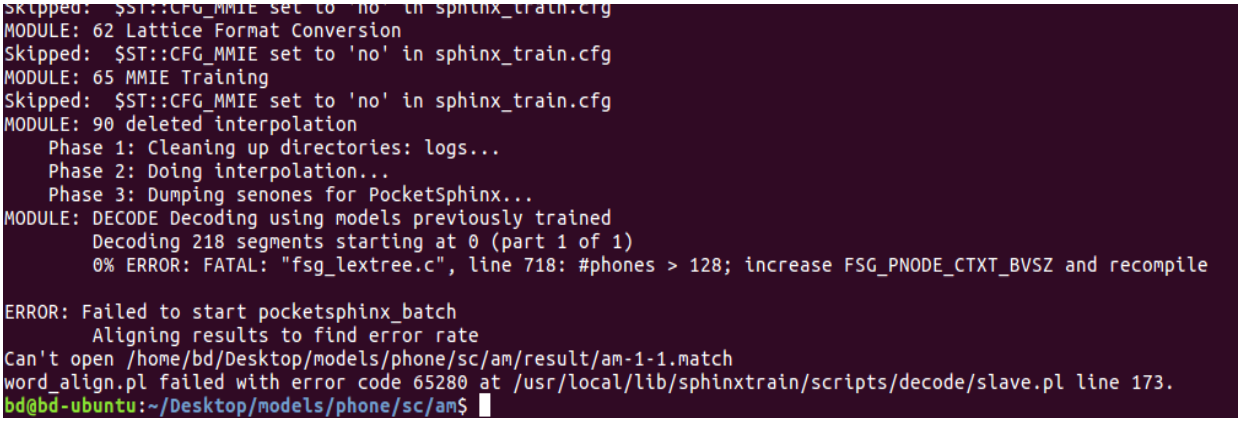


Figure 4.3 The error message at the end of training the WDP models and while trying to decode these models.

Consequently, it is found out necessary to study the source code of the Pocketsphinx, and correct the problem by increasing the maximum number of phones that Pocketsphinx support. Hence, in this research, we have studied the source code of Pocketsphinx, increased the allowed maximum number of phones in *fsg_lextree.h* file and then re-compiled both Pocketsphinx and Pocketsphinx-Android to apply the changes. By doing this, the WDP models are decoded and their recognition accuracy calculated successfully, and also the models are run on both the virtual emulator and the physical mobile phones smoothly.

4.3 Testing the Models

S. Furui [57] states, speech recognition systems can be evaluated using subjective or objective methods. The former directly involve human subjects during measurement, whereas the latter, typically uses prerecorded speech, and thus do not directly involve human subjects. Accordingly, in this research work both of these evaluation techniques are employed, the first technique on the first performance evaluation which is Category I and the second evaluation technique on the second performance evaluation which is Category II.

The evaluations in Category I, as mentioned previously, are carried out to determine which of the three model types that CMUSphinx support perform best on each of the two models, WDP and WDCVS, and if major recognition accuracy differences are observed to select the ones with best accuracies and to incorporate them into the prototype application for the tests on mobile phone. Performance evaluations in the second category, Category II, on the other hand are carried out on

a mobile phone, to decide the best of the WDP and WDCVS models trained with the three model type and then to decide whether the WDP or WDCVS model has the best recognition performance.

According to D. Jurafsky and J. H. Martin [17], the standard evaluation metric for speech recognition systems is the word error rate. The word error rate is based on how much the word string returned by the recognizer differs from a correct or reference transcription.

Word error rate is can be defined as: $WER = \frac{S+D+I}{N}$

Where, *S* is the number of substitutions, *D* is the number of the deletions, *I* is the number of the insertions and *N* is the number of words in the reference.

Hence, in all the performance evaluation activities in the two categories, WER is used to evaluate recognition accuracy of each model.

4.3.1 Performance Evaluation: Category – I

This evaluation as indicated above is, the first category of performance evaluation and aims at investigating the effect of the three model types on the two models and then determining which of the WDP model trained with three model types scored the best recognition accuracy, and also which of the WDCVS model trained with three model types scored the best recognition accuracy. Hence, performance evaluations in Category I are performed by executing the *sphinxtrain -s decode run* command on the Linux shell, where the path of the directory containing the created models and the feature files of the test set are specified.

From the recorded speech data containing 1,218 sentences, around 82% (1,000 sentences) of the data is used for training, and the remaining 18% (218 sentences, containing 1,251 words) is used for testing. Hence, the 1,251 words, which are composed of all the 36 words, are used in computing the WERs of the trained models.

The performance evaluation is started with the WDP model; therefore, three tests are done to get the WERs of the three model types applied on the WDP model. Table 4.3 below presents their recognition accuracies.

Model Type	Errors	WER (%)	Accuracy (%)
continuous	6	0.48	99.52
semi-continuous	6	0.48	99.52
PTM	15	1.20	98.80

Table 4.3 Recognition accuracies of the WDP model trained with the three model types

In like manner, other three tests are conducted to get the recognition performance of the WDCVS trained with the three model types. Table 4.4 below presents the results of the tests.

Model Type	Errors	WER (%)	Accuracy
continuous	6	0.48	99.52
semi-continuous	6	0.48	99.52
PTM	12	0.96	99.04

Table 4.4 Recognition accuracies of the WDCVS model trained with the three model types

From the test results presented in the two tables above, both the WDP and also WDCVS models trained with continuous and semi-continuous model types have scored the best recognition accuracies, which is also equal. Each of these four models have recognized 6 words incorrectly, and scored an equal recognition accuracy of 99.52%. On the other hand, the two models, WDP and WDCVS, trained with PTM model types have wrongly recognized 12 and 15 words and scored a lower recognition accuracy of 98.80% and 99.04% respectively.

As it can be clearly seen that, both of the models have achieved excellent recognition accuracies, whichever model type is used, and therefore, this shows the suitability of using the WDP and WDCVS models for small vocabulary ASR systems. It is also evident that, these accuracy figures are obtained by training the models with the recommended approximate values of number of senones and number of densities presented in Table 4.2, and therefore, these recognition accuracies can still be improved if further experiments conducted with different values of these and other parameters.

Screenshots taken during the decoding of the WDCVS model trained with continuous and PTM and WDP model trained with semi-continuous, are shown below in Figures 4.4, 4.5 and 4.6 below, respectively. The aligned reference and hypothesis sentences, generated during the decoding phase of WDP model trained with continuous model type is presented under Appendix 7.

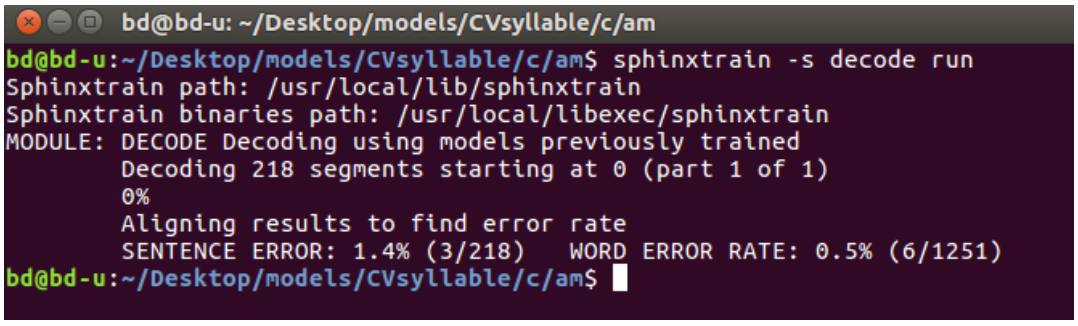


Figure 4.4 Screenshot of decoding result of the WDCVS trained with continuous

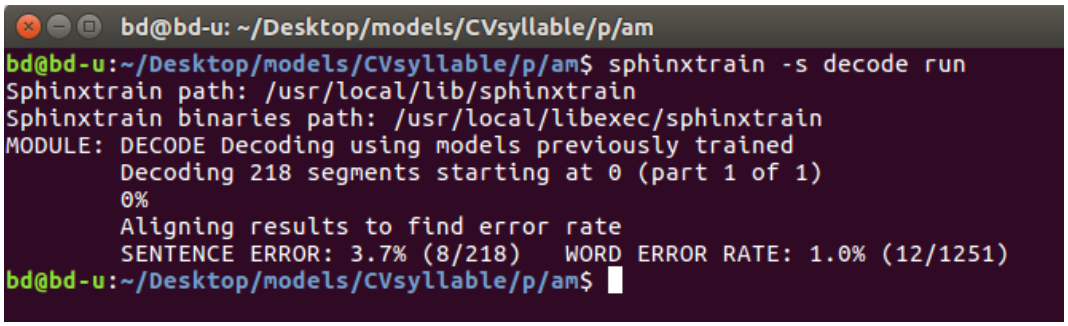


Figure 4.5 Screenshot of decoding result of the WDCVS trained with PTM model type

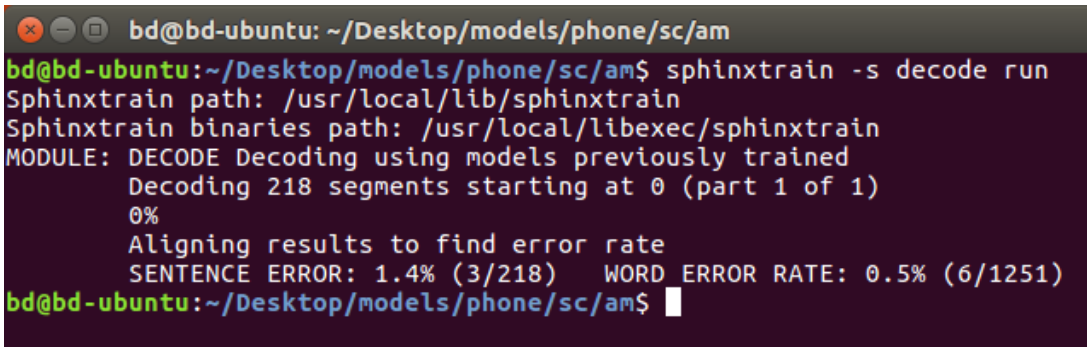


Figure 4.6 Screenshot of decoding result of the WDP trained with semi-continuous

As discussed above, because all the six experiments yielded very good recognition accuracies and no significant variations are observed among the recognition accuracies, we decided to conduct the next category of performance evaluation, which is Category II, on a mobile phone using all the six models by incorporating each model into the prototype application.

4.3.2 Performance Evaluation: Category – II

In this category of performance evaluation, the evaluation is done on an actual physical mobile phone, after each model is incorporated into the prototype application and after the application is installed on a mobile phone. Unlike the performance evaluations in Category I, here the evaluations are conducted not only to determine and select the best of the WDP and the WDCVS model trained with the three model types, but also aims at determining whether WDP or WDCVS is the best for implementing a connected word ASR system on mobile phones.

Before the prototype application is installed on a mobile phone, it is tested on a virtual emulator phone (as shown in Fig. 3.7) to insure its functionality on the computer where some of the speech corpus is collected, all the training activities are carried out and where the prototype application itself is developed. And, this is important to handle and deal with if any technical and operational issues will be encountered. The microphone used in this preliminary test is the same headset used to record the audio data on computer.

The three grammar files combined in the training and decoding phases above, are separated back into three: menu, commands and digits, since the prototype application first chooses the type of utterance need to be recognized i.e. digit or command, and because it uses the grammar files for this separately.

After installing the prototype application on a mobile phone, all the six acoustic models are tested turn by turn using the prototype application. Unlike the performance evaluations carried out in the first category, where the evaluations are conducted by running commands on the trained models, here, all the utterances that the recognizer recognizes are spoken to the prototype application. And for this, the same person's voice, whose speech is used for training the models, is used in this performance evaluation phase also. Similarly, the environment where the evaluations are carried out is the same with that of the environment where the speech corpus is collected.

Though, subjective measures include level of intelligibility, general impression, annoyance, user-friendliness, intuitiveness and the likes, in this research work, subjective evaluation technique is used to only evaluate the recognition accuracy of the models or the prototype application on mobile phones [57].

The prototype Amharic ASR for mobile phones developed, is designed to recognize all the digit words 0 through 100, 12 command phrases and also 2 command phrases i.e. "በድምጽ መደወል" and "በድምጽ ማዘዝ", for switching between the phone dialing and commanding activities, making a total of 115 utterances (23 single and 92 connected words) that can be recognized by the prototype application. And these 115 words and phrases are consisted of 207 words formed by all the 36 words used in this study. Thus, in this category of performance evaluation, all the 115 utterances are used in testing the performance of each of the six models and the 207 words are used to calculate the WER of each model.

The testing is performed subjectively, by first listing all the 115 words and phrases sequentially and then by speaking each utterance to the prototype application and registering all correctly recognized words as correct and all incorrectly recognized words i.e. substituted, deleted or inserted, as error. And finally, the WER of each model is computed using these numbers. Table 4.5 below presents a summary of the recognition accuracies of the two models trained with the three model types, together with the words recognized incorrectly.

Model	Model type	Errors	WER (%)	Accuracy (%)
WDP	Continuous	6	2.89	97.11
	semi-continuous	8	3.86	96.14
	PTM	5	2.42	97.58
WDCVS	Continuous	4	1.93	98.07
	semi-continuous	5	2.42	97.58
	PTM	5	2.42	97.58

Table 4.5 Recognition accuracies of all the models on mobile phone

As it can be seen from Table 4.5 above, the accuracies obtained in this testing, which is conducted on mobile phones, have shown minimum decrease in recognition accuracy from that of the recognition accuracies obtained when testing the models using the test set. However, despite the accuracy reductions, all the models have still achieved an excellent recognition accuracy. In this category of tests, the best recognition accuracies for the two models is found to be 97.58% for WDP trained with PTM and 98.07% for WDCVS trained with continuous model type.

In general, it can be concluded that, WDCVS model particularly a WDCVS model trained with a continuous model type is the best to be used in deploying an Amharic connected word, speaker dependent, small vocabulary speech recognition system that can be used to command and control mobile phones in an offline mode; though, WDP model is also a possible alternative to be considered. Screenshots taken during this performance evaluation phase are shown in Fig. 4.7 below.

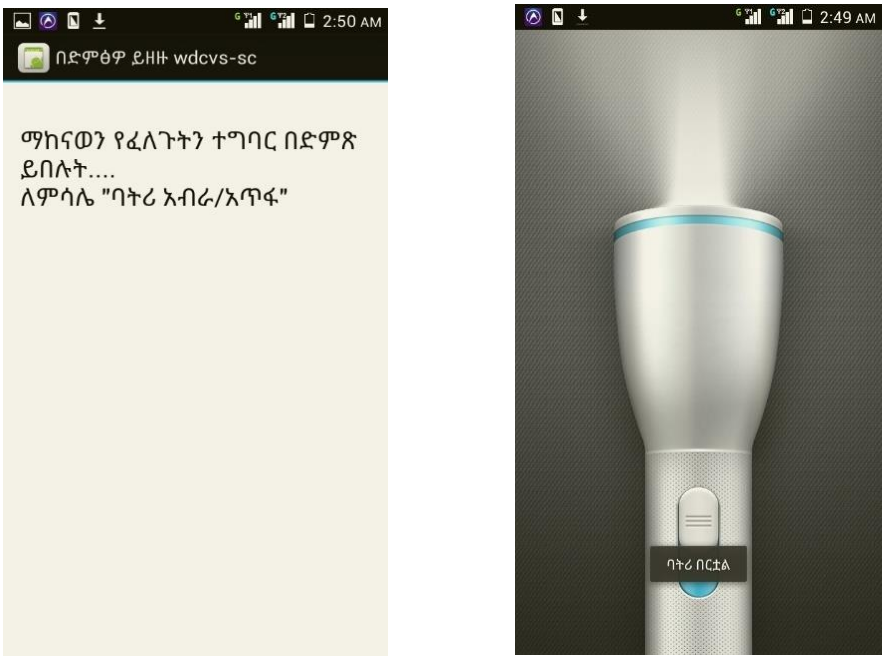


Figure 4.7 Screenshots of the Prototype Amharic ASR system for mobile phones, installed on a physical mobile phone: Huawei G730-U00 mobile phone

In addition, although recognition speed and storage space are not considered in this research work, both of which place significant constraints on the use of ASR systems on mobile phones, the two models have showed a very fast recognition speed and also an acceptable file size. For instance, the file sizes of each of the two best modes are 1.79 MB and 5.66 MB, for WDCVS model trained with continuous and WDP model trained with PTM, respectively.

Chapter Five

Conclusion and Recommendation

5.1 Conclusion

The primary objective of this research work is to explore ways of creating and integrating a connected word, offline, speaker-dependent, small vocabulary Amharic WDP and WDCVS based ASR system, that can be used to command and control mobile phones using Amharic language. Previous research works in the area of Amharic speech recognition, have identified different gaps and explored different methods to narrow and fill the gaps that exist in the area. Besides the different methods and techniques, different recognition units i.e. phones, triphones [32] , CV syllables [7], [28], [32], whole words [44], are also used by the researchers as a means to achieve their objectives. In like manner, in this research work, a new method towards recognition unit, that is the use of WDP and WDCVS, for acoustic modeling is explored with a primary goal of investigating way of creating and integrating an improved Amharic connected word ASR system with mobile phones. For this, a total of 36 words are used in both the models to recognize the Amharic digits from 0 to 100 and a limited set of command and control words/phrases spoken in a connected manner.

To model the acceptable sequence of words in any given context three distinct JSGF grammar files are created and used, the first for the digit words, the second for the command words and the third to switch between the two sets of words since it is possible to switch between grammars in CMU Pocketsphinx at the time of recognition.

In this research work, two categories of performance evaluations are carried out. In the first category, six tests are conducted to see the effect the three model types, i.e. continuous, semi-continuous and PTM have on each of the two models, i.e. WDCVS and WDP, and then select the WDCVS and WDP models with best recognition accuracies and then test their performance on a mobile phone. These tests the WDP and WDCVS models trained with continuous and semi-continuous model types have scored the best (which is also equal) recognition of 99.52%. And the WDP and WDCVS, trained with PTM model types have scored a lower recognition accuracy of 98.80% and 99.04% respectively.

Because no significant recognition accuracy differences observed among all the six models, all the models are incorporated with the prototype application and the second category of performance evaluation is carried out on a mobile phone with the aim of deciding the best of the WDP and WDCVS models trained with the three model types and then which of the two models have scored the best accuracy on mobile phones.

In light of this, performance evaluations conducted on mobile phones using the prototype application have yielded the best recognition accuracy of 98.07% WDCVS trained with continuous model type and 97.58% WDP trained with PTM. Thus, the WDCVS model trained with continuous model type is found to be the best for developing an Amharic ASR system on mobile phones, given the speech corpus used in this study, and the recommended number of senones and densities used to train the models.

From the challenges in conducting this study, lack of adequate literatures in the area, lack of a reference speech corpus, and some technical challenges like the one we encountered while the training and decoding of the WDP model, were the main ones.

In general, this study has demonstrated the integration and also usability of an enhanced connected word Amharic ASR system that can be used in commanding and controlling mobile phones using the two models; and therefore, it can be concluded that the primary objective of the research is fully achieved.

5.2 Recommendation

Research and developments in the area of ASR system for mobile phones for local language including Amharic language is almost untouched when compared to the ever-growing prevalence of mobile phones, and therefore demands more focus. Generally, based on the findings of the study, the following recommendations are formulated as a future research direction in the area of Amharic ASR systems for mobile phones:

- The development of a speech recognition based application for mobile phones, an area which has not been covered in this particular study, can be investigated and the findings of this study can be used, to fully deploy an Amharic ASR system on mobile phones.

- This study investigated the use of speaker dependent ASR system form mobiles, hence, speaker independent system can be investigated by training the system with a large speech corpus representing voices of various people.

- The development of ready-made corpus or reference database for researchers, is also an area worth considering. A good corpus contains speech samples from significantly large number of speakers with different demographic and socio-linguistic groups. Here, since an ASR system on mobile phones are usually used in different environment and context from that of PCs, particular emphasis need to be given to have a representative speech corpus.

- Even though, there are different pronunciation for some of the words used in this particular research work i.e. "አስራ አራት" for "አስራራት" and "አስራምስት" for "አስራ አምስት", these pronunciation variations are not taken into account and it would be better to incorporate this feature as it would have an impact on the recognition accuracy.

- To train a model properly, different values of the different parameters, i.e. number of senones and the number of Gaussian mixtures need to experimented, which also are not considered in this particular research work. Hence, we recommend future works in this area to consider investigating the effects of these parameters, and select the ones which result in the best accuracies for the particular development set used, as the optimal numbers depend the development set.

References

[1] L. Rabiner and B. H. Juang, Fundamentals of Speech Recognition, Engelwood NJ: Prentice-Hall Inc, 1993.

[2] Cyra Aleeza A. Ebuenga, Hazel Joy V. Catacutan, Christine Dianne Z. Castro and Prof. Albert A. Vinluan, "Javawiki : A Mobile-Based Java Programming Language Dictionary Using Speech Recognition," May, 2015.

[3] Z. H. Tan and B. Lindberg, Automatic Speech Recognition on Mobile Devices and over Communication Networks, London: Springer-Verlag London Limited, 2008.

[4] ITU and G3ict, "Making mobile phones and services accessible for persons with disabilities," International Telecommunication Union, G3ict – The global initiative for inclusive ICTs, 2012.

[5] W. Meisel, "“Life on-the-Go”: The Role of Speech Technology in Mobile Applications," in *Advances in Speech Recognition*, A. Neustein, Ed., New York, Springer, 2010.

[6] L. Rabiner and B. H. Juang, Fundamentals of Speech Recognition, Engelwood NJ: Prentice-Hall Inc, 1993, p. 8.

[7] Solomon Birhanu, "Isolated Amharic Consonant-Vowel syllable recognition," *MSc. Thesis, Addis Ababa University School of Information Studies for Africa,* 2001.

[8] Tinbit Kassahun, "Automatic Amharic Speech Recognition to command mobile devices," *MSc. Thesis, Computer Science, School of Computer Science & Technology, HiLCoE,* Aug 2015.

[9] Mikhael Aweke, "Automatic Amharic Speech Recognition for Phone dialing using Hidden Markov Model (HMM)," *MSc Thesis, Computer Science, Addis Ababa University, College of Natural Sciences,* Oct 2015.

[10] M. Phillips, J. Nguyen and A. Mischke, "“Why Tap When You Can Talk?”: Designing Multimodal Interfaces for Mobile Devices that Are Effective, Adaptive and Satisfying to the User," in *Advances in Speech Recognition*, A. Neustein, Ed., New York, Springer, 2010.

[11] Solomon Tefera (PhD), Martha Yifru (PhD) and Ermias Abebe, "Towards the Development of Speech Recognition Application," Addis Ababa, 2015.

[12] K. Chinetha, J. Daphney Joann and A. Shalini, "An Evolution of Android Operating System and Its Version," *International Journal of Engineering and Applied Sciences (IJEAS),* vol. 2, no. 2, 2015.

[13] B. H. Juang and L. Rabiner, "Hidden Markov Models for Speech Recognition," *TECHNOMETRICS,* vol. 33, no. 3, pp. 251-272, 1991.

[14] David Huggins-Daines et. al., "Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System For Hand-Held Devices," in *International Conference on Conference Paper in Acoustics, Speech, and Signal Processing*, 2006.

[15] CMUSphinx, "CMUSphinx Tutorial For Developers," CMUSphinx, 12 Sep 2017. [Online]. Available: https://cmusphinx.github.io/wiki/tutorial/. [Accessed 23 Aug 2017].

[16] X. Huang, A. Acero and H. W. Hon, Spoken Language Processing: A Guide to Theory, Algorithm, and System Development, New Jersey: Prentice Hall, 2001.

[17] D. Jurafsky and J. H. Martin, Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall, 2006.

[18] X. Huang and L. Deng, "An Overview of Modern Speech Recognition," in *Handbook of Natural Language Processing*, 2nd ed., Cambridge, UK, Taylor and Francis Group, LLC, 2010.

[19] F. Jelinek, Statistical Methods for Speech Recognition, London: The MIT Press, 1997.

[20] J. Baker, Stochastic modeling for automatic speech recognition, in D. R. Reddy, (ed.), Speech Recognition, New York: Academic Press, 1975.

[21] F. Jelinek, "Continuous speech recognition by statistical methods," *IEEE,* vol. 64, no. 4, p. 532–557, 1976.

[22] S. K. Saksamudre et. al., "A Review on Different Approaches for Speech Recognition System," *International Journal of Computer Applications,* vol. 115, no. 22, pp. 23-28, 2015.

[23] S. Karpagavalli and E. Chandra, "A Review on Automatic Speech Recognition Architecture and Approaches," *International Journal of Signal Processing, Image Processing and Pattern Recognition,* vol. 9, no. 4, pp. 393-404, 2016.

[24] M. A. Anusuya and S. Katti, "Speech Recognition by Machine: A Review," *International Journal of Computer Science and Information Security,* vol. 6, no. 3, 2009.

[25] S. Karpagavalli and E. Chandra, "A Review on Sub-word unit Modeling in Automatic Speech Recognition," *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP),* vol. 6, no. 6, pp. 77-84, 2016.

[26] A. Waibel and K.-F. Lee, Readings in Speech Recognition, San Mateo, California: Morgan Kaufmann Publishers, Inc., 1990.

[27] Y. Chow et al., "Context-dependent modeling for acoustic phonetic recognition of continuous speech," *Proc. IEEE Int. Conference on Acoustic Speech Signal Processing,* 1985.

[28] Solomon Tefera and W. Menzel, "Syllable-Based Speech Recognition for Amharic," Prague, 2007.

[29] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of The IEEE,* vol. 77, no. 2, 1989.

[30] Tolemariam Fufa, A Typology of Verbal Derivation in Ethiopian Afro-Asiatic Languages, Janskerkhof: LOT, 2009.

[31] V. Clark, A. Eschholz and A. Rosa, Language: Introductory Readings, New York: St. Martin's Press, 1985.

[32] Kinfe Tadesse, "Sub-word based Amharic speech recognizer: An experiment using Hidden Markov Model (HMM)," *MSc Thesis, School of Information Studies for Africa, Addis Ababa University,* June 2002.

[33] B. Gambäck and E. Samuel, "Classifying Amharic News Text Using Self-organizing Maps," Michigan, 2005.

[34] M. Bender, Language in Ethiopia, London: Oxford University Press, 1976.

[35] Million Meshesha and C. V. Jawahar, "Indigenous Scripts Of African Languages," *Indilinga,* vol. 6, no. 2, pp. 132-142, May 2015.

[36] Sintayehu Hirphasa, "Designing an Information Extraction System for Amharic Vacancy Announcement Text," *MSc. Thesis, Addis Ababa University, School of Graduate Studies, School of Information Science,* JUNE 2013.

[37] Rev. C. W. Isenberg, Grammar Of The Amharic Language, London: Printed for the church missionary society, 1842.

[38] Z. H. Tan and B. Lindberg, "Speech Recognition on Mobile Devices," in *Mobile Multimedia Processing Fundamentals, Methods, and Applications*, Berlin Heidelberg, Springer-Verlag, 2010.

[39] Mayorga P. et. al., "Audio packet loss over IP and speech recognition. In ,," Virgin Islands, 2003.

[40] D. Pearce, "Robustness to transmission channel — The DSR approach," Norwich, UK, 2004.

[41] Z. H. Tan, P. Dalsgaard and L. B. , "Automatic speech recognition over errorprone wireless networks," *Speech Communication,* vol. 47, no. 1, p. 220–242, 2005.

[42] B. Delaney, "Reduced Energy Consumption and Improved Accuracy for Distributed Speech Recognition in Wireless Environments," *Ph.D. Thesis, Georgia Institute of Technology, School of Electrical and Computer Engineering,* Sep. 2004.

[43] B. Suhm, B. Myers and A. Waibel, "Multimodal Error Correction for Speech User Interfaces," *ACM Transactions on Computer-Human Interaction,* vol. 8, no. 1, p. 60–98, 2001.

[44] Martha Yifru, "Application of Amharic Speech Recognition System to Command and Control Computer: An Experiment with Microsoft Word," *MSc Thesis, Addis Ababa University, School of Graduate Studies,* July 2003.

[45] H. Hugeng and E. Hansel, "Implementation of Android Based Speech Recognition for Indonesian Geography Dictionary," *ULTIMA Computing,* vol. 7, no. 2, pp. 76-82, 2015.

[46] N. Mulhern et.al., "Designing Android Applications with both Online and Offline Voice Control of Household Devices," 2013.

[47] Liuxinfei and Zhouhui, "A Chinese Small Vocabulary Offline Speech Recognition System Based on Pocketsphinx in Android Platform," *Applied Mechanics and Materials,* vol. 623, pp. 267-273, 2014.

[48] C. Vimala and V. Radha, "A Review on Speech Recognition Challenges and Approaches," *World of Computer Science and Information Technology Journal (WCSIT),* vol. 2, no. 1, pp. 1-7, 2012.

[49] S. D. Dhingra, G. Nijhawan and Poonam, "Isolated Speech Recognition Using MFCC and DTW," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* vol. 2, no. 8, pp. 4085-4092, 2013.

[50] Sebsibe H/Mariam et al., "Unit Selection Voice for Amharic Using FESTVOX," in *In proceeding of the 5th ISCA Speech Synthesis Workshop*, Pittsburgh, 2004.

[51] Martha Yifiru Ph.D, Solomon Teferra Abate Ph.D and B. Laurent Ph.D, "Using different acoustic, lexical and language modeling units for ASR of an under-resourced language – Amharic," *Speech Communication,* vol. 56, pp. 181-194, 2014.

[52] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the CMU-cambridge toolkit," *5th European Conference on Speech Communication and Technology,* p. 2707– 2710, 1997.

[53] Andrew Hunt, Speech Works International, "JSpeech Grammar Format," Sun Microsystems, Inc., 05 June 2000. [Online]. Available: http://www.w3.org/TR/jsgf. [Accessed 13 2 2017].

[54] N. V. Shmyrev, "github.com," GitHub, Inc., 28 Apr 2014. [Online]. Available: https://github.com/cmusphinx/pocketsphinx-android-demo. [Accessed 16 3 2017].

[55] N. V. Shmyrev, "github.com," GitHub, Inc., 14 Apr 2014. [Online]. Available: https://github.com/cmusphinx/pocketsphinx-android. [Accessed 23 8 2017].

[56] P. Gilski and J. Stefanski, "Android OS: A Review," *TEM,* vol. 4, no. 1, pp. 116-120, 2015.

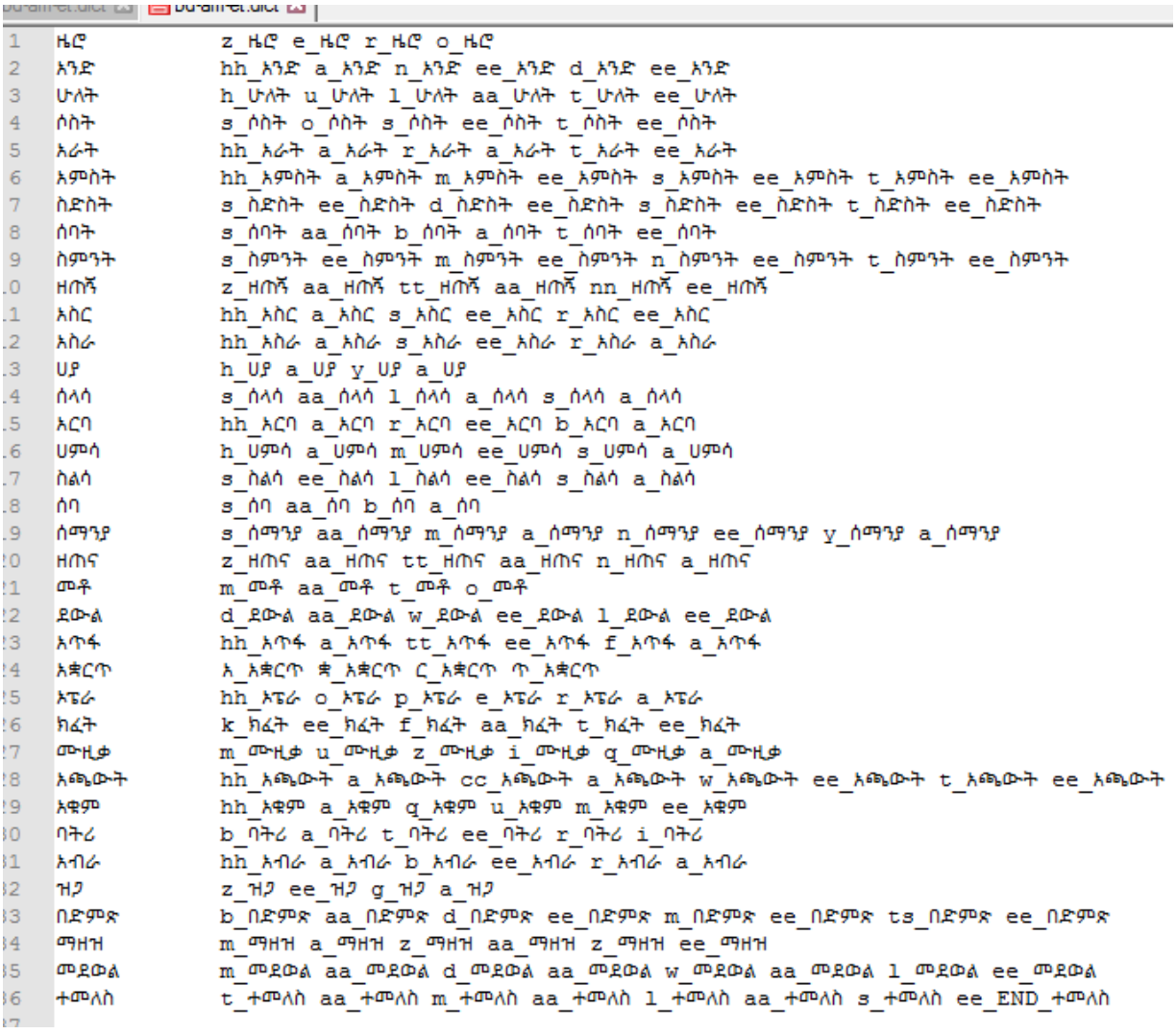
[57] S. Furui, "SPEECH AND SPEAKER RECOGNITION EVALUATION," in *Evaluation of Text and Speech Systems*, Tokyo, Japan, Springer, 2007, pp. 1-27.

[58] K. Tadesse, "Sub-word based Amharic speech recognizer: An experiment using Hidden Markov Model (HMM)," *MSc Thesis, School of Information Studies for Africa, Addis Ababa University, Ethiopia,* 2002.

Appendixes

Appendix 1

Appendix 1-A Word-dependent phone dictionary



1	ዜሮ	z_ዜሮ e_ዜሮ r_ዜሮ o_ዜሮ
2	አንድ	hh_አንድ a_አንድ n_አንድ ee_አንድ d_አንድ ee_አንድ
3	ሁለት	h_ሁለት u_ሁለት l_ሁለት aa_ሁለት t_ሁለት ee_ሁለት
4	ሶስት	s_ሶስት o_ሶስት s_ሶስት ee_ሶስት t_ሶስት ee_ሶስት
5	አራት	hh_አራት a_አራት r_አራት a_አራት t_አራት ee_አራት
6	አምስት	hh_አምስት a_አምስት m_አምስት ee_አምስት s_አምስት ee_አምስት t_አምስት ee_አምስት
7	ስድስት	s_ስድስት ee_ስድስት d_ስድስት ee_ስድስት s_ስድስት ee_ስድስት t_ስድስት ee_ስድስት
8	ሰባት	s_ሰባት aa_ሰባት b_ሰባት a_ሰባት t_ሰባት ee_ሰባት
9	ስምንት	s_ስምንት ee_ስምንት m_ስምንት ee_ስምንት n_ስምንት ee_ስምንት t_ስምንት ee_ስምንት
10	ዘጠኝ	z_ዘጠኝ aa_ዘጠኝ tt_ዘጠኝ aa_ዘጠኝ nn_ዘጠኝ ee_ዘጠኝ
11	አስር	hh_አስር a_አስር s_አስር ee_አስር r_አስር ee_አስር
12	አስራ	hh_አስራ a_አስራ s_አስራ ee_አስራ r_አስራ a_አስራ
13	ሀያ	h_ሀያ a_ሀያ y_ሀያ a_ሀያ
14	ሰላሳ	s_ሰላሳ aa_ሰላሳ l_ሰላሳ a_ሰላሳ s_ሰላሳ a_ሰላሳ
15	አርባ	hh_አርባ a_አርባ r_አርባ ee_አርባ b_አርባ a_አርባ
16	ሀምሳ	h_ሀምሳ a_ሀምሳ m_ሀምሳ ee_ሀምሳ s_ሀምሳ a_ሀምሳ
17	ስልሳ	s_ስልሳ ee_ስልሳ l_ስልሳ ee_ስልሳ s_ስልሳ a_ስልሳ
18	ሰባ	s_ሰባ aa_ሰባ b_ሰባ a_ሰባ
19	ሰማንያ	s_ሰማንያ aa_ሰማንያ m_ሰማንያ a_ሰማንያ n_ሰማንያ ee_ሰማንያ y_ሰማንያ a_ሰማንያ
20	ዘጠና	z_ዘጠና aa_ዘጠና tt_ዘጠና aa_ዘጠና n_ዘጠና a_ዘጠና
21	መቶ	m_መቶ aa_መቶ t_መቶ o_መቶ
22	ደውል	d_ደውል aa_ደውል w_ደውል ee_ደውል l_ደውል ee_ደውል
23	አጥፋ	hh_አጥፋ a_አጥፋ tt_አጥፋ ee_አጥፋ f_አጥፋ a_አጥፋ
24	አቋርጥ	ኣ_አቋርጥ ቋ_አቋርጥ ር_አቋርጥ ጥ_አቋርጥ
25	ኦፔራ	hh_ኦፔራ o_ኦፔራ p_ኦፔራ e_ኦፔራ r_ኦፔራ a_ኦፔራ
26	ክፈት	k_ክፈት ee_ክፈት f_ክፈት aa_ክፈት t_ክፈት ee_ክፈት
27	ሙዚቃ	m_ሙዚቃ u_ሙዚቃ z_ሙዚቃ i_ሙዚቃ q_ሙዚቃ a_ሙዚቃ
28	አጫውት	hh_አጫውት a_አጫውት cc_አጫውት a_አጫውት w_አጫውት ee_አጫውት t_አጫውት ee_አጫውት
29	አቁም	hh_አቁም a_አቁም q_አቁም u_አቁም m_አቁም ee_አቁም
30	ባትሪ	b_ባትሪ a_ባትሪ t_ባትሪ ee_ባትሪ r_ባትሪ i_ባትሪ
31	አብራ	hh_አብራ a_አብራ b_አብራ ee_አብራ r_አብራ a_አብራ
32	ዝጋ	z_ዝጋ ee_ዝጋ g_ዝጋ a_ዝጋ
33	በድምጽ	b_በድምጽ aa_በድምጽ d_በድምጽ ee_በድምጽ m_በድምጽ ee_በድምጽ ts_በድምጽ ee_በድምጽ
34	ማዘዝ	m_ማዘዝ a_ማዘዝ z_ማዘዝ aa_ማዘዝ z_ማዘዝ ee_ማዘዝ
35	መደወል	m_መደወል aa_መደወል d_መደወል aa_መደወል w_መደወል aa_መደወል l_መደወል ee_መደወል
36	ተመለስ	t_ተመለስ aa_ተመለስ m_ተመለስ aa_ተመለስ l_ተመለስ aa_ተመለስ s_ተመለስ ee_END_ተመለስ
37		

Appendix 1-B, Word-dependent CV syllable dictionary

1	ዜሮ	ዜ_ዜሮ ሮ_ዜሮ
2	አንድ	አ_አንድ ን_አንድ ድ_አንድ
3	ሁለት	ሁ_ሁለት ለ_ሁለት ት_ሁለት
4	ሶስት	ሶ_ሶስት ስ_ሶስት ት_ሶስት
5	አራት	አ_አራት ራ_አራት ት_አራት
6	አምስት	አ_አምስት ም_አምስት ስ_አምስት ት_አምስት
7	ስድስት	ስ_ስድስት ድ_ስድስት ስ_ስድስት ት_ስድስት
8	ሰባት	ሰ_ሰባት ባ_ሰባት ት_ሰባት
9	ስምንት	ስ_ስምንት ም_ስምንት ን_ስምንት ት_ስምንት
10	ዘጠኝ	ዘ_ዘጠኝ ጠ_ዘጠኝ ኝ_ዘጠኝ
11	አስር	አ_አስር ስ_አስር ር_አስር
12	አስራ	አ_አስራ ስ_አስራ ራ_አስራ
13	ሀያ	ሀ_ሀያ ያ_ሀያ
14	ሰላሳ	ሰ_ሰላሳ ላ_ሰላሳ ሳ_ሰላሳ
15	አርባ	አ_አርባ ር_አርባ ባ_አርባ
16	ሀምሳ	ሀ_ሀምሳ ም_ሀምሳ ሳ_ሀምሳ
17	ስልሳ	ስ_ስልሳ ል_ስልሳ ሳ_ስልሳ
18	ሰባ	ሰ_ሰባ ባ_ሰባ
19	ሰማንያ	ሰ_ሰማንያ ማ_ሰማንያ ን_ሰማንያ ያ_ሰማንያ
20	ዘጠና	ዘ_ዘጠና ጠ_ዘጠና ና_ዘጠና
21	መቶ	መ_መቶ ቶ_መቶ
22	ይውል	ደ_ይውል ው_ይውል ል_ይውል
23	አጥፋ	አ_አጥፋ ጥ_አጥፋ ፋ_አጥፋ
24	አቋርጥ	አ_አቋርጥ ቋ_አቋርጥ ር_አቋርጥ ጥ_አቋርጥ
25	አፔራ	አ_አፔራ ፔ_አፔራ ራ_አፔራ
26	ክፈት	ክ_ክፈት ፈ_ክፈት ት_ክፈት
27	ሙዚቃ	ሙ_ሙዚቃ ዚ_ሙዚቃ ቃ_ሙዚቃ
28	አጫውት	አ_አጫውት ጫ_አጫውት ው_አጫውት ት_አጫውት
29	አቁም	አ_አቁም ቁ_አቁም ም_አቁም
30	ባትሪ	ባ_ባትሪ ት_ባትሪ ሪ_ባትሪ
31	አብራ	አ_አብራ ብ_አብራ ራ_አብራ
32	ዝጋ	ዝ_ዝጋ ጋ_ዝጋ
33	በድምጽ	በ_በድምጽ ድ_በድምጽ ም_በድምጽ ጽ_በድምጽ
34	ማዘዝ	ማ_ማዘዝ ዘ_ማዘዝ ዝ_ማዘዝ
35	መደወል	መ_መደወል ደ_መደወል ወ_መደወል ል_መደወል
36	ተመለስ	ተ_ተመለስ መ_ተመለስ ለ_ተመለስ ስ_END_ተመለስ
37		

Appendix 2

Appendix 2, Amharic Alphabets

Appendix 3

Appendix 3, Sample transcription sentences used for training the models

am_train.transcription | am_test.transcription

```
<s> ዘጠና አንድ ዘጠና ሁለት ዘጠና ሶስት ዘጠና አራት ዘጠና አምስት </s> (m027-1-100)
<s> ስድስት ሰባት ስምንት ዘጠኝ አስር </s> (m030-1-100)
<s> አስራ አንድ አስራ ሁለት አስራ ሶስት አስራ አራት አስራ አምስት </s> (m031-1-100)
<s> አስራ ስድስት አስራ ሰባት አስራ ስምንት አስራ ዘጠኝ ሀያ </s> (m032-1-100)
<s> ሀያ አንድ ሀያ ሁለት ሀያ ሶስት ሀያ አራት ሀያ አምስት </s> (m033-1-100)
<s> ሀያ ስድስት ሀያ ሰባት ሀያ ስምንት ሀያ ዘጠኝ ሰላሳ </s> (m034-1-100)
<s> ሰላሳ አንድ ሰላሳ ሁለት ሰላሳ ሶስት ሰላሳ አራት ሰላሳ አምስት </s> (m035-1-100)
<s> ሰላሳ ስድስት ሰላሳ ሰባት ሰላሳ ስምንት ሰላሳ ዘጠኝ አርባ </s> (m036-1-100)
<s> አርባ አንድ አርባ ሁለት አርባ ሶስት አርባ አራት አርባ አምስት </s> (m037-1-100)
<s> አርባ ስድስት አርባ ሰባት አርባ ስምንት አርባ ዘጠኝ ሀምሳ </s> (m038-1-100)
<s> ሀምሳ አንድ ሀምሳ ሁለት ሀምሳ ሶስት ሀምሳ አራት ሀምሳ አምስት </s> (m039-1-100)
<s> ሀምሳ ስድስት ሀምሳ ሰባት ሀምሳ ስምንት ሀምሳ ዘጠኝ ስልሳ </s> (m040-1-100)
<s> ስልሳ አንድ ስልሳ ሁለት ስልሳ ሶስት ስልሳ አራት ስልሳ አምስት </s> (m041-1-100)
<s> ሰባ ስድስት ሰባ ሰባት ሰባ ስምንት ሰባ ዘጠኝ ሰማንያ </s> (m044-1-100)
<s> ሰማንያ አንድ ሰማንያ ሁለት ሰማንያ ሶስት ሰማንያ አራት ሰማንያ አምስት </s> (m045-1-100)
<s> ሰማንያ ስድስት ሰማንያ ሰባት ሰማንያ ስምንት ሰማንያ ዘጠኝ ዘጠና </s> (m046-1-100)
<s> ዘጠና አንድ ዘጠና ሁለት ዘጠና ሶስት ዘጠና አራት ዘጠና አምስት </s> (m047-1-100)
<s> ዘጠና ስድስት ዘጠና ሰባት ዘጠና ስምንት ዘጠና ዘጠኝ መቶ </s> (m048-1-100)
<s> አንድ ሁለት ሶስት አራት አምስት </s> (m049-1-100)
<s> ስድስት ሰባት ስምንት ዘጠኝ አስር </s> (m050-1-100)
<s> አስራ አንድ አስራ ሁለት አስራ ሶስት አስራ አራት አስራ አምስት </s> (m051-1-100)
<s> ሀያ ስድስት ሀያ ሰባት ሀያ ስምንት ሀያ ዘጠኝ ሰላሳ </s> (m054-1-100)
<s> ሰላሳ አንድ ሰላሳ ሁለት ሰላሳ ሶስት ሰላሳ አራት ሰላሳ አምስት </s> (m055-1-100)
<s> ሰላሳ ስድስት ሰላሳ ሰባት ሰላሳ ስምንት ሰላሳ ዘጠኝ አርባ </s> (m056-1-100)
<s> አርባ አንድ አርባ ሁለት አርባ ሶስት አርባ አራት አርባ አምስት </s> (m057-1-100)
<s> አርባ ስድስት አርባ ሰባት አርባ ስምንት አርባ ዘጠኝ ሀምሳ </s> (m058-1-100)
<s> ሀምሳ አንድ ሀምሳ ሁለት ሀምሳ ሶስት ሀምሳ አራት ሀምሳ አምስት </s> (m059-1-100)
<s> ሀምሳ ስድስት ሀምሳ ሰባት ሀምሳ ስምንት ሀምሳ ዘጠኝ ስልሳ </s> (m060-1-100)
<s> ስልሳ አንድ ስልሳ ሁለት ስልሳ ሶስት ስልሳ አራት ስልሳ አምስት </s> (m061-1-100)
<s> ስልሳ ስድስት ስልሳ ሰባት ስልሳ ስምንት ስልሳ ዘጠኝ ሰባ </s> (m062-1-100)
<s> ሰባ አንድ ሰባ ሁለት ሰባ ሶስት ሰባ አራት ሰባ አምስት </s> (m063-1-100)
<s> ሰባ ስድስት ሰባ ሰባት ሰባ ስምንት ሰባ ዘጠኝ ሰማንያ </s> (m064-1-100)
<s> ሰማንያ አንድ ሰማንያ ሁለት ሰማንያ ሶስት ሰማንያ አራት ሰማንያ አምስት </s> (m065-1-100)
<s> ሰማንያ ስድስት ሰማንያ ሰባት ሰማንያ ስምንት ሰማንያ ዘጠኝ ዘጠና </s> (m066-1-100)
<s> ዘጠና አንድ ዘጠና ሁለት ዘጠና ሶስት ዘጠና አራት ዘጠና አምስት </s> (m067-1-100)
<s> ዘጠና ስድስት ዘጠና ሰባት ዘጠና ስምንት ዘጠና ዘጠኝ መቶ </s> (m068-1-100)
<s> አንድ አንድ አንድ </s> (m069-1)
```

Appendix 4

Appendix 4 – The Phonset used in developing the WDCVS model

SIL
ዜ_ዜሮ
ሮ_ዜሮ
አ_አንድ
ን_አንድ
ድ_አንድ
ሁ_ሁለት
ለ_ሁለት
ት_ሁለት
ሶ_ሶስት
ስ_ሶስት
ት_ሶስት
አ_አራት
ራ_አራት
ት_አራት
አ_አምስት
ም_አምስት
ስ_አምስት
ት_አምስት
ስ_ስድስት
ድ_ስድስት
ት_ስድስት
ሰ_ሰባት
ባ_ሰባት
ት_ሰባት
ስ_ስምንት
ም_ስምንት
ን_ስምንት
ት_ስምንት
ዘ_ዘጠኝ
ጠ_ዘጠኝ
ኝ_ዘጠኝ
አ_አስር
ስ_አስር
ር_አስር
አ_አስራ
ስ_አስራ
ራ_አስራ
ሀ_ሀያ
ያ_ሀያ
ሰ_ሰላሳ
ላ_ሰላሳ
ሳ_ሰላሳ
አ_አርባ
ር_አርባ
ባ_አርባ
ሀ_ሀምሳ
ም_ሀምሳ
ሳ_ሀምሳ
ስ_ስልሳ
ል_ስልሳ
ሳ_ስልሳ
ሰ_ሰባ
ባ_ሰባ
ሰ_ሰማንያ
ማ_ሰማንያ
ን_ሰማንያ
ያ_ሰማንያ
ዘ_ዘጠና
ጠ_ዘጠና
ና_ዘጠና
መ_መቶ
ቶ_መቶ
ደ_ደውል
ው_ደውል
ል_ደውል
አ_አጥፋ
ጥ_አጥፋ
ፋ_አጥፋ
አ_አቋርጥ
ቋ_አቋርጥ
ር_አቋርጥ
ጥ_አቋርጥ
አ_አፔራ
ፔ_አፔራ
ራ_አፔራ
ክ_ክፈት
ፈ_ክፈት
ት_ክፈት
ሙ_ሙዚቃ
ዚ_ሙዚቃ
ቃ_ሙዚቃ
አ_አጫውት
ጫ_አጫውት
ው_አጫውት
ት_አጫውት
አ_አቁም
ቁ_አቁም
ም_አቁም
ባ_ባትሪ
ት_ባትሪ
ሪ_ባትሪ
አ_አብራ
ብ_አብራ
ራ_አብራ
አ_አጥፋ
ጥ_አጥፋ
ፋ_አጥፋ
ዝ_ዝጋ
ጋ_ዝጋ
በ_በድምጽ
ድ_በድምጽ
ም_በድምጽ
ጽ_በድምጽ
ማ_ማዘዝ
ዘ_ማዘዝ
ዝ_ማዘዝ
መ_መደወል
ደ_መደወል
ወ_መደወል
ል_መደወል
ተ_ተመለስ
መ_ተመለስ
ለ_ተመለስ
ስ_END_ተመለስ

Appendix 5

Appendix 5, Sample of the code used in the development of the prototype application

```
public class PocketSphinxActivity extends Activity implements
        RecognitionListener {

    /* Named searches allow to quickly reconfigure the decoder */
    private static final String KWS_SEARCH = "wakeup";
    private static final String DIGITS_SEARCH = "በድምጽ መደወል";
    private static final String PHONE_SEARCH = "በድምጽ ማዘዝ";
    private static final String MENU_SEARCH = "menu";

    int STATES ;

    /* Used to handle permission request */
    private static final int PERMISSIONS_REQUEST_RECORD_AUDIO = 1;

    private SpeechRecognizer recognizer;
    private HashMap<String, Integer> captions;

    @Override
    public void onCreate(Bundle state) {
        super.onCreate(state);

        // Prepare the data for UI
        captions = new HashMap<String, Integer>();
        captions.put(MENU_SEARCH, R.string.menu_caption);
        captions.put(DIGITS_SEARCH, R.string.digits_caption);
        captions.put(PHONE_SEARCH, R.string.phone_caption);
        setContentView(R.layout.main);
        ((TextView) findViewById(R.id.caption_text))
                .setText("እያዘጋጅ ነው....");
        // Check if user has given permission to record audio
        int permissionCheck =
ContextCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.RECORD_AUDIO);
        if (permissionCheck != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.RECORD_AUDIO}, PERMISSIONS_REQUEST_RECORD_AUDIO);
            return;
        }
        runRecognizerSetup();
    }
    private void runRecognizerSetup() {
        // Recognizer initialization is a time-consuming and it involves IO,
        // so we execute it in async task
        new AsyncTask<Void, Void, Exception>() {
            @Override
            protected Exception doInBackground(Void... params) {
                try {
                    Assets assets = new Assets(PocketSphinxActivity.this);
                    File assetDir = assets.syncAssets();
                    setupRecognizer(assetDir);
                } catch (IOException e) {
                    return e;
                }
                return null;
            }
```

Appendix 6

Appendix 6, Decision tree built during training the WDP with continuous model type

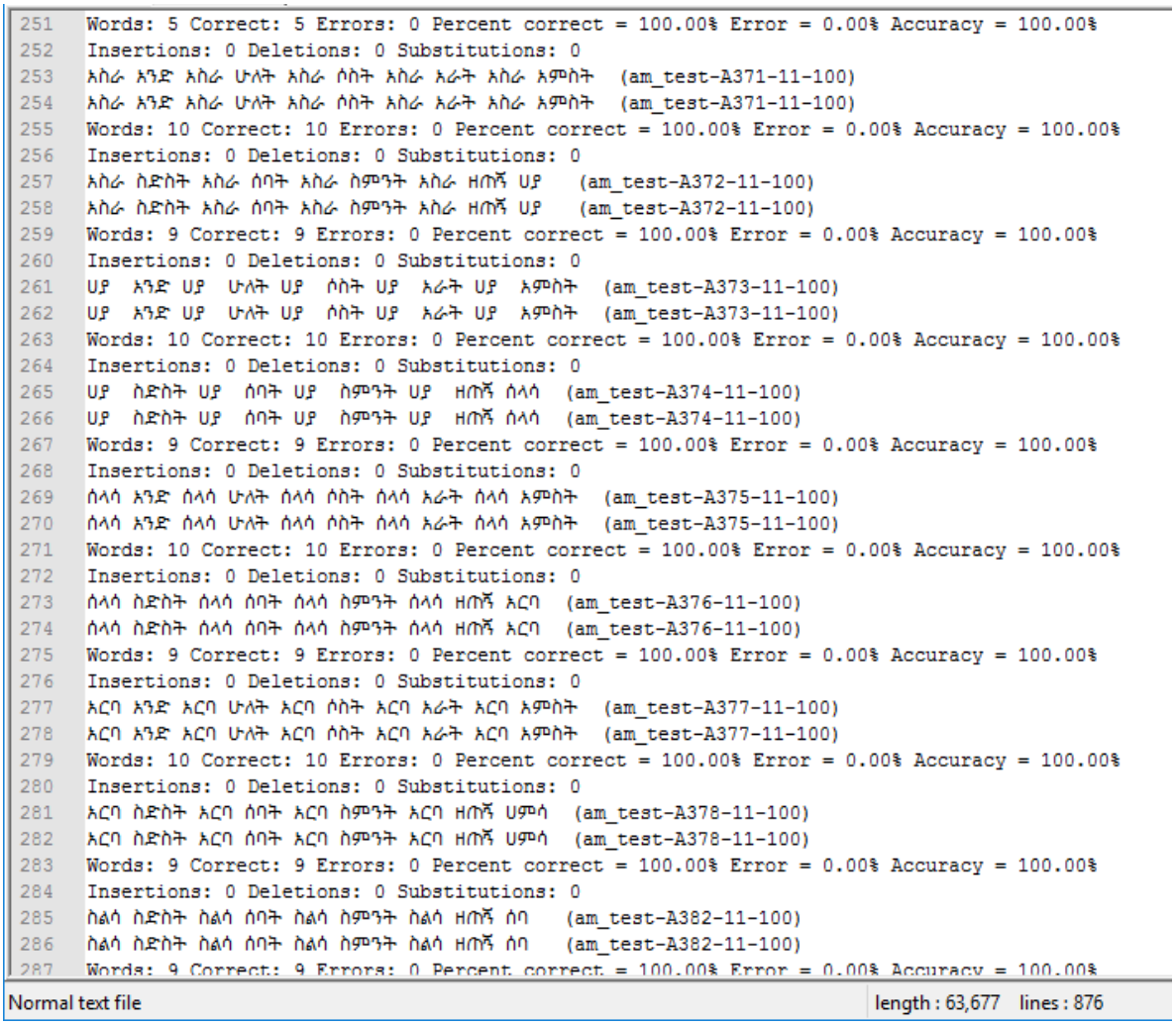
```
Current configuration:
[NAME]                                  [DEFLT]  [VALUE]
-allphones                              no no
-cntthresh                              0.00001     1.000000e-05
-csplitmax                              100         2000
-csplitmin                              1   1
-csplitthr                              8e-4        0.000000e+00
-example                                no          no
-fullvar                                no          no
-help                                   no          no
-meanfn
                                        /home/bd/Desktop/m2/am2/c/am/model_parameters/am.cd_c
ont_untied/means
-mixwfn
                                        /home/bd/Desktop/m2/am2/c/am/model_parameters/am.cd_c
ont_untied/mixture_weights
-moddeffn
                                        /home/bd/Desktop/m2/am2/c/am/model_architecture/am.unti
ed.mdef
-mwfloor                                1e-4        1.000000e-08
-phone                                              a_ዘጠና
-psetfn
                                        /home/bd/Desktop/m2/am2/c/am/model_architecture/am.tree
_questions
-ssplitmax                              5   7
-ssplitmin                              1   1
-ssplitthr                              8e-4        0.000000e+00
-state                                              0
-stwt                                               1.0,0.05,0.0,
-treefn
                                        /home/bd/Desktop/m2/am2/c/am/trees/am.unpruned/a_ዘጠና-
0.dtree
-ts2cbfn                                .semi.      .cont.
-varfloor                               0.00001     1.000000e-05
-varfn
                                        /home/bd/Desktop/m2/am2/c/am/model_parameters/am.cd_c
ont_untied/variances

INFO: main.c(223): Reading: /home/bd/Desktop/m2/am2/c/am/model_architecture/am.untied.mdef
INFO: model_def_io.c(573): Model definition info:
INFO: model_def_io.c(574): 825 total models defined (183 base, 642 tri)
INFO: model_def_io.c(575): 3300 total states
INFO: model_def_io.c(576): 2475 total tied states
INFO: model_def_io.c(577): 549 total tied CI states
INFO: model_def_io.c(578): 183 total tied transition matrices
INFO: model_def_io.c(579): 4 max state/model
INFO: model_def_io.c(580): 4 min state/model
INFO: main.c(242): Building trees for [a_ዘጠና n_ዘጠና SIL e] through [a_ዘጠና n_ዘጠና z_ዘጠኝ e]
```

```
INFO: main.c(269): Covering states |[918 950]| == 33
INFO: main.c(275): Reading:
/home/bd/Desktop/m2/am2/c/am/model_parameters/am.cd_cont_untied/mixture_weights
INFO: s3mixw_io.c(173): Read
/home/bd/Desktop/m2/am2/c/am/model_parameters/am.cd_cont_untied/mixture_weights [33x1x1 array]
INFO: main.c(311): 11 of 11 models have observation count greater than 0.000010
INFO: main.c(95): nrm stwt: 0.952 0.048 0.000
INFO: s3gau_io.c(169): Read /home/bd/Desktop/m2/am2/c/am/model_parameters/am.cd_cont_untied/means
[2475x1x1 array]
INFO: s3gau_io.c(169): Read /home/bd/Desktop/m2/am2/c/am/model_parameters/am.cd_cont_untied/variances
[2475x1x1 array]
INFO: main.c(549): Reading: /home/bd/Desktop/m2/am2/c/am/model_architecture/am.tree_questions
INFO: main.c(588): 172 total simple questions (164 phone; 8 word bndry)
INFO: main.c(590): 40 Left Only questions, and 40 Right Only questions
INFO: dtree.c(1439): stop. leaf nodes are specific
INFO: dtree.c(1478): Final simple tree
|( QUESTION0_18_R 1 -3.475e+05 2.319e+03 1.831e+03
|    ( QUESTION4 1 -7.054e+03 1.456e+02 3.528e+01
|       ( - -6.222e+03 5 0)
|       ( - -6.863e+02 1 0))
|    ( QUESTION0_17_R 1 -3.381e+05 9.820e+02 1.796e+03
|       ( QUESTION2 1 -2.662e+03 1.815e+02 1.398e+01
|          ( - -1.444e+03 1 0)
|          ( - -1.036e+03 1 0))
|       ( SILENCE 1 -3.345e+05 2.068e+02 1.782e+03
|          ( - -8.396e+04 1 0)
|          ( - -2.503e+05 2 0))))
s>    ( QUESTION0_18_R 1 -3.475e+05 2.319e+03 1.831e+03
s>       ( QUESTION4 1 -7.054e+03 1.456e+02 3.528e+01
s>          ( - -6.222e+03 5 0)
s>          ( - -6.863e+02 1 0))
s>       ( QUESTION0_17_R 1 -3.381e+05 9.820e+02 1.796e+03
s>          ( QUESTION2 1 -2.662e+03 1.815e+02 1.398e+01
s>             ( - -1.444e+03 1 0)
s>             ( - -1.036e+03 1 0))
s>          ( SILENCE 1 -3.345e+05 2.068e+02 1.782e+03
s>             ( - -8.396e+04 1 0)
s>             ( - -2.503e+05 2 0))))INFO: dtree.c(1315): Comp split 0
INFO: dtree.c(1439): stop. leaf nodes are specific
INFO: dtree.c(1478): Final simple tree
|( SILENCE 1 -3.345e+05 2.068e+02 1.782e+03
|    ( - -8.396e+04 1 0)
|    ( - -2.503e+05 2 0))
s>    ( SILENCE 1 -3.345e+05 2.068e+02 1.782e+03
s>       ( - -8.396e+04 1 0)
s>       ( - -2.503e+05 2 0))INFO: dtree.c(1439): stop. leaf nodes are specific
INFO: dtree.c(1478): Final simple tree
|( QUESTION2 1 -9.803e+03 1.484e+02 4.926e+01
|    ( - -1.444e+03 1 0)
|    ( QUESTION4 1 -8.210e+03 1.230e+02 4.128e+01
|       ( - -6.222e+03 5 0)
|       ( QUESTION0_13_R 1 -1.865e+03 1.428e+02 1.000e+01
|          ( - -1.036e+03 1 0)
```

```
|          ( - -6.863e+02 1 0))))
s>   ( QUESTION2 1 -9.803e+03 1.484e+02 4.926e+01
s>      ( - -1.444e+03 1 0)
s>      ( QUESTION4 1 -8.210e+03 1.230e+02 4.128e+01
s>         ( - -6.222e+03 5 0)
s>         ( QUESTION0_13_R 1 -1.865e+03 1.428e+02 1.000e+01
s>            ( - -1.036e+03 1 0)
s>            ( - -6.863e+02 1 0))))INFO: dtree.c(1315): Comp split 1
INFO: dtree.c(1315): Comp split 2
INFO: dtree.c(1439): stop. leaf nodes are specific
INFO: dtree.c(1478): Final simple tree
|( QUESTION0_13_R 1 -7.375e+03 1.170e+02 3.728e+01
|    ( - -1.036e+03 1 0)
|    ( - -6.222e+03 5 0))
s>   ( QUESTION0_13_R 1 -7.375e+03 1.170e+02 3.728e+01
s>      ( - -1.036e+03 1 0)
s>      ( - -6.222e+03 5 0))INFO: dtree.c(1439): stop. leaf nodes are specific
INFO: dtree.c(1478): Final simple tree
|( QUESTION0_13_R 1 -2.278e+03 1.473e+02 1.198e+01
|    ( - -1.444e+03 1 0)
|    ( - -6.863e+02 1 0))
s>   ( QUESTION0_13_R 1 -2.278e+03 1.473e+02 1.198e+01
s>      ( - -1.444e+03 1 0)
s>      ( - -6.863e+02 1 0))INFO: dtree.c(1315): Comp split 3
INFO: dtree.c(1315): Comp split 4
INFO: dtree.c(1315): Comp split 5
INFO: dtree.c(1318): stop. leaf nodes are specific
Wed Sep 27 06:21:48 2017
```

Appendix 7

Appendix 7, Aligned reference and hypothesis sentences generated during the decoding the WDP model trained with continuous model type



```
251  Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
252  Insertions: 0 Deletions: 0 Substitutions: 0
253  አስራ አንድ አስራ ሁለት አስራ ሶስት አስራ አራት አስራ አምስት  (am_test-A371-11-100)
254  አስራ አንድ አስራ ሁለት አስራ ሶስት አስራ አራት አስራ አምስት  (am_test-A371-11-100)
255  Words: 10 Correct: 10 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
256  Insertions: 0 Deletions: 0 Substitutions: 0
257  አስራ ስድስት አስራ ሰባት አስራ ስምንት አስራ ዘጠኝ ሀያ   (am_test-A372-11-100)
258  አስራ ስድስት አስራ ሰባት አስራ ስምንት አስራ ዘጠኝ ሀያ   (am_test-A372-11-100)
259  Words: 9 Correct: 9 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
260  Insertions: 0 Deletions: 0 Substitutions: 0
261  ሀያ  አንድ ሀያ  ሁለት ሀያ  ሶስት ሀያ  አራት ሀያ  አምስት  (am_test-A373-11-100)
262  ሀያ  አንድ ሀያ  ሁለት ሀያ  ሶስት ሀያ  አራት ሀያ  አምስት  (am_test-A373-11-100)
263  Words: 10 Correct: 10 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
264  Insertions: 0 Deletions: 0 Substitutions: 0
265  ሀያ  ስድስት ሀያ  ሰባት ሀያ  ስምንት ሀያ  ዘጠኝ ሰላሳ  (am_test-A374-11-100)
266  ሀያ  ስድስት ሀያ  ሰባት ሀያ  ስምንት ሀያ  ዘጠኝ ሰላሳ  (am_test-A374-11-100)
267  Words: 9 Correct: 9 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
268  Insertions: 0 Deletions: 0 Substitutions: 0
269  ሰላሳ አንድ ሰላሳ ሁለት ሰላሳ ሶስት ሰላሳ አራት ሰላሳ አምስት  (am_test-A375-11-100)
270  ሰላሳ አንድ ሰላሳ ሁለት ሰላሳ ሶስት ሰላሳ አራት ሰላሳ አምስት  (am_test-A375-11-100)
271  Words: 10 Correct: 10 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
272  Insertions: 0 Deletions: 0 Substitutions: 0
273  ሰላሳ ስድስት ሰላሳ ሰባት ሰላሳ ስምንት ሰላሳ ዘጠኝ አርባ  (am_test-A376-11-100)
274  ሰላሳ ስድስት ሰላሳ ሰባት ሰላሳ ስምንት ሰላሳ ዘጠኝ አርባ  (am_test-A376-11-100)
275  Words: 9 Correct: 9 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
276  Insertions: 0 Deletions: 0 Substitutions: 0
277  አርባ አንድ አርባ ሁለት አርባ ሶስት አርባ አራት አርባ አምስት  (am_test-A377-11-100)
278  አርባ አንድ አርባ ሁለት አርባ ሶስት አርባ አራት አርባ አምስት  (am_test-A377-11-100)
279  Words: 10 Correct: 10 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
280  Insertions: 0 Deletions: 0 Substitutions: 0
281  አርባ ስድስት አርባ ሰባት አርባ ስምንት አርባ ዘጠኝ ሀምሳ  (am_test-A378-11-100)
282  አርባ ስድስት አርባ ሰባት አርባ ስምንት አርባ ዘጠኝ ሀምሳ  (am_test-A378-11-100)
283  Words: 9 Correct: 9 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
284  Insertions: 0 Deletions: 0 Substitutions: 0
285  ስልሳ ስድስት ስልሳ ሰባት ስልሳ ስምንት ስልሳ ዘጠኝ ሰባ   (am_test-A382-11-100)
286  ስልሳ ስድስት ስልሳ ሰባት ስልሳ ስምንት ስልሳ ዘጠኝ ሰባ   (am_test-A382-11-100)
287  Words: 9 Correct: 9 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
Normal text file                                          length : 63,677   lines : 876
```

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

_____

Bewunetu Dagne

October, 2017

Confirmation

Thesis has been submitted for examination with our approval as University advisor.

_____

Martha Yifru (PhD)

October, 2017