Addis Ababa University

College of Natural and Computational Science

School of Information Science

Amharic Speech Search Using Text Word Query

By

GETNET MEZGEBU

Jan 20, 2022

Addis Ababa, Ethiopia

Addis Ababa University

College of Natural and Computational Science

School of Information Science

Amharic Speech Search Using Text Word Query

A Thesis Submitted to the School of Information Science of Addis Ababa University in Partial Fulfillment of the Requirement for the Degree of Master of Science in Information Science

By: GETNET MEZGEBU

Advisor: Dr. SOLOMON TEFERRA

Jan 20, 2022

Addis Ababa, Ethiopia

Addis Ababa University

College of Natural and Computational Science

School of Information Science

Amharic Speech Search Using Text Word Query

By: GETNET MEZGEBU

Name and signature of Members of the Examining Board

Solomon Teferra (PhD)	_____	_____
Advisor	Signature	Date
_____(PhD)	_____	_____
Examiner	Signature	Date
_____(PhD)	_____	_____
Examiner	Signature	Date

Declaration

This thesis has not previously been accepted for any degree and is not being concurrently submitted in candidature for any degree in any university.

I declare that the thesis is a result of my investigation, except where otherwise stated. I have undertaken the study independently with the guidance and support of my research advisor. Other sources are acknowledged by citations giving explicit references. A list of references is appended.

Signature: _____

Getnet Mezgebu

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

Advisor's Signature: _____

Solomon Teferra (PhD)

Acknowledgment

First and foremost, I would like to thank almighty God for helping in every aspect of my journey throughout my life despite so many difficulties.

Next, I would like to thank my advisor, Dr. Solomon, who helped me in shaping my research topic and study from the beginning to the end. I sincerely appreciate his kindness to help me whenever help was needed and his availability to motivate and encourage me throughout the study period with his critical comment and suggestions.

Finally, I would like to thank my family and friends for their inspiration and support throughout my life.

Table of Contents

Abstract

In a world where more than 7000 languages are spoken and the processing and storage power of machines are maximized, many speech data are produced every day using different languages. Amharic is one of the languages spoken in the East African country, Ethiopia. Searching for a particular spoken word with its respective time frame inside a given audio file is a challenge.

The main objective of this research was to investigate the development of a system that can search speech and locate utterance with its respective time from the Amharic audio file by using Amharic text word query.

In this study, the researchers followed an experimental research methodology. To meet the research objective, we have conducted an experiment to get the optimal segmentation for which we can achieve the lowest WER of the ASR system that decodes the segmented speech. We have also experimented on the use of previously developed speech corpus, which is in a broadcast domain, together with the in-domain speech corpus, which is the Bible domain; we have developed for our research. The performance of the ASR obtained by combining the two different domains shows a better WER than that of using only LVCSR. On the other hand, the comparison of automatically segmented speech with automatic sentence-like segmentations shows closer WER with the manually (by hand) segmented speech. Using the optimal automatic segmentation and LVCSR, the researchers developed a text-based STD which can locate the time interval upon which the query term is located. The text-based STD was developed with ASR having a WER of 53% and 46 % using LVCSR and by combining LVCSR and Bible speech respectively. The developed STD has a Graphical User Interface (GUI) which will make searching easy to use and friendly. We found that the performance of ASR affects the performance of STD since not all terms are fully transcribed.

Keywords: Speech segmentation, Spoken Term Detection, Automatic Speech Recognition, Manual Speech Segmentation

List of Tables

List of Figures

List of Acronyms

ASR	Automatic Speech Recognition
ATWV	Actual weighted Term Value
CMU	Carnegie Mellon University
HMM	Hidden Markov Model
IR	Information Retrieval
KWS	Keyword Spotting
LM	Language Model
LVCSR	Large Vocabulary Continuous Speech Recognizer
OOV	Out-Of –Vocabulary
QBE STD	Query By Example Spoken Term Detection
SCR	Spoken Content Retrieval
SoS	Search on Speech
STD	Spoken Term Detection
WER	Word Error Rat

CHAPTER ONE

INTRODUCTION

This chapter describes the overall background and organization of the thesis, statement of the problem, research questions and objectives, the methodology employed, and significance along with the scope and limitation of the study.

1.1. Background

Audio files are generated from different sources like the internet and social media by individuals and organizations. Moreover, recent technological development in storage space of machines and their affordable prices make the production and storage of audio files less challenging. Audio files can now be easily recorded and shared using different mediums or platforms to instantly reach the public.

With around 7000 languages spoken in the world [**1**], audio files are available in different languages. Amharic is the official working language of the government of Ethiopia; an East African country with a population of over 100 million. Audio data, referring to all audible data in the frequency range of 20–20,000 Hz, are abundantly found in Amharic language via the various private, social and government platforms. The audio files are played using audio player software and social Media like you-tube. Audio players do not have any functionality of searching a particular word in a speech. However, the general public and organizations use these audio files to analyze the required information and satisfy their information needs. In doing so, searching for particular spoken speech saves time and increases effectiveness.

Audio data can be searched by the file name, title, and speech and by using audio search engines which will retrieve relevant documents as per the users' need from document collections such as the internet. However, individuals and organizations are also interested in searching for a particular spoken word from specific audio file as we are searching a word in a document file.

In accessing information of interest from an audio file, there should be an effective and efficient mechanism to search for a particular spoken word [**2**]. This would avoid users from wasting their time by listening to the whole audio speech.

Audio files can be searched based on the audio type (music, audio and speech), learning (supervised and unsupervised), query representation, Keyword Spotting (KWS), text based

Spoken Term Detection (STD) and Query By Example Spoken Term Detection (QBE STD). To search for a particular spoken word, the contents of the audio data have to be identified. Spoken Content Retrieval (SCR) is the task of returning speech media results that are relevant to an information need expressed as a user query. SCR focuses on meaning-based relevance ranking [**3**]. On the other hand, STD focuses on finding the query term inside the audio file which also extends to locating the time stamp of the query term. This task requires the implementation of Automatic Speech Recognition (ASR) [**4**].

The output of this research work can became an input for other related research so that search engines developed for Amharic multimedia retrieve relevant files with the respective time frame. It further enables to highlight the words and locate their time frames within the audio files.

To date, no research has been done specifically on searching user's text query from Amharic audio file. In the absence of such research, all stakeholders like individuals, groups, and organizations continue having difficulty in gathering relevant information from audio data. Research was also needed to assess the effect of using either manually or automatically segmented speech on the ASR performance. The present study was therefore designed to undertake a comparative study (manually versus automatically segmented speech) and research on searching for a particular word in an audio file.

1.2. Motivation

The availability of huge amount of information stored in audio and video repositories around the world increasing interest in Search on Speech (SoS) [**5**] which focuses on retrieving speech content from audio repositories that matches user queries; i.e., searching the audio/speech by either using any term of interest by text or segment of the audio or voice. In addition individuals and organization wants to search a particular spoken word from a given audio file that can be found in social medias and any audio file whose content is in Amharic. Therefore there should be some way that would allow users locate the exact location of the spoken utterance or word without having to listen to the whole audio file from Amharic audio file.

1.3. Statement of the Problem and Justification

The high prevalence of social media and multimedia in our interconnected global society today created the need to access different audio files. Individuals and organizations use this audio file to satisfy their information needs. They may know that the speaker spoke a word but may not know in which part of the audio file that word is spoken. Automatically locating the part of the audio file where a particular word is spoken is challenging. Different researchers have conducted several research in different languages. Since every language has its own feature, the solution to speech search is language specific.

Individuals and organizations use the multimedia files to satisfy their information needs. This need will be satisfied if there is a way that will help them to effectively use those information within the audio files. Users get information through listening the audio file. However, recently available audio players and social Medias do not allow users to search for a particular Amharic spoken word from the audio file. If users wants to locate a particular spoken word from a given audio file, they have to listen the whole audio file from the beginning to the end or have to drag by estimating at which part of the audio the spoken word is located. For example, to find for a spoken word ኢትዮጵያ /Ethiopia that was spoken in a given audio file having a length n in time, users might listen to the whole audio file or guess for the spoken word (ኢትዮጵያ) within that audio file. However, for organizations and individuals especially for those who need to analyze the audio file, listening for the whole audio file to obtain a particular spoken word is a challenge.

Amharic audio files are based on contents which are in Amharic language. Languages which are spoken in Ethiopia including Amharic do have a distinguishing characteristic from other languages such as English. The existence of glottal, palatal, and labialized consonants makes the Amharic language different from other languages. In addition, Amharic is one of the inflated languages [**6**]. Searching for a particular spoken word from a given audio file is a challenge for languages that are spoken in Ethiopia: particularly Amharic which has distinguishing characteristics or property. Consequently, it is not possible to directly use models which are implemented for other languages including English.

Currently, there are online web applications that allow users to convert a given audio file to an English text. In addition, the study by K. Kodlekere et.al [**7**] on Keyword Based Indexing of a Multimedia File in English language allows users to search for a particular spoken speech using text and display the time frame and the utterance. However, no research has been done on

searching or locating the time frame interval of the spoken word (utterance) from the audio file in any Ethiopian languages.

Previous research towards the development of an Amharic audio search engine [**8**] focused on how to retrieve relevant audio documents as per the users query. On the other hand, research conducted on the development of a search engine for other languages showed capability for searching the audio and video content by converting the audio/video file in to a simplified time-marked stream of text [**9**].

With the increase interst on SoS , Spoken Term Detection (STD) a type of SoS that helps to retrieve speech data through using text as a query word that represents a particular speech utterance [**10**]. Besides this, the study on searching for speech remains on its infancy [**11**].

The development of such a system requires us to develop at least three components: speech segmentation, speech recognition and text-based search engine unless we develop a system that can search directly on the speech signal. The development of these components is language-dependent. Currently, even if there are applications which are developed by using ASR, performance improvement is needed for tasks such as speech search as noted by Yifiru et al. [**12**].

In researching the development of a system that can accept a text word query from a user, we need to investigate the development of the three components. 1) The development of an optimal speech segmentation to be recognized by the ASR component; 2) To investigate the improvement of the ASR performance on automatically segmented, which may have wrong segmentation, speech; and 3) To investigate the development of a search engine that searches users query in recognized files, which may have recognition errors.

At the end of the present study, the following research questions were aimed to be answered.

1. What is the effect of manually and automatically segmented test speech on ASR performance?
2. What is the effect of ASR recognition errors on searching the recognized text?
3. What is the effect of ASR recognition errors on searching using different domains?

1.4. Objective

1.4.1. General Objective

The general objective of this research is to investigate the development of a system that can search speech utterance with its respective time interval from the Amharic audio file by using Amharic query text.

1.4.2. Specific Objective

- To undertake literature search on the topic and have a thorough understanding of the problem as well as getting an insight on the methods used to solve a problem.
- To explore audio searching techniques and applying the suitable searching technique for Amharic speech audio data.
- To prepare training and test speech (automatically and manually segmented speech) corpus for ASR development.
- To develop ASR using automatically (prepared by ourselves) and manually (adopted from previous research work) segmented training speech corpus.
- To compare the performance of the developed ASR systems on automatically segmented and manually segmented test speech.
- To develop Large Vocabulary Continuous Speech Recognizer (LVCSR) acoustic model by adopting already available training speech corpus and by combining the LVCSR with Bible speech corpus.
- To integrate the ASR systems with java (indexing or alignment).
- To develop text-based STD that will search and locate time of the query word.
- To evaluate the performance of text-based STD.

1.5. Methodology

In order to conduct this particular research work, different techniques and methods were used to meet the above-mentioned objective.

This thesis work includes the development of two ASR systems. The first ASR system was used for the comparison of automatically and manually segmented speeches. This recognition system is developed using a Bible speech corpus that we collected from you-tube and respective

transcription text from online web resources. In addition to the web resources, we also used manually segmented Bible speech from the research work done by Mekonen [**13**]. This allowed ASR comparison using manually versus automatically segmented speech.

The second recognition system was used in the development of text-based STD. The system was developed by adopting the LVCSR news corpus prepared by Abate et. al. [**14**].

1.5.1. Segmentation

Speech segmentation is a process of decomposing a long speech signal into shorter length [**15**]. Python's pydub package will be used for automatic speech segmentation [**16**]. This package will segment the audio file on every frame using silence.

1.5.2. Searching

Searching for a particular speech from a given audio file can be done using different techniques such as STD, QBE STD and keyword Detection [**10**]. Since our main research interest is on searching a particular spoken speech from a given audio file by using Amharic text as a query word, we used the text-based STD as a searching technique.

1.5.3. Development Tools and Techniques

There are lots of ASR development toolkits. Out of which in this particular thesis work we used CMU Sphinx which is an open-source ASR development toolkit. Sphinx4 (which is a pure Java speech recognition library that provides a quick and easy API to convert the speech recordings into text with the help of CMU Sphinx acoustic models) [**17**] . This library/API can easily be accessed by java which is selected as the most popular object-oriented programing language in 2020 [**18**] . Using java, the Sphinx4 and pythons pydub package we integrated different components. The basic and intermediate result of the integration was the task of alignment which is the component of text-based STD that could be easily searched using query.

1.6. Significance of the Study / Expected Benefits

- Since there is no prior research made on the effect of automatically segmented test speech on the performance of ASR on Amharic language, the finding of this research adds knowledge in understanding the effect of automatically and manually tests speech recognition of ASR.
- The result of this research will enhance multimedia information retrieval. Since Multimedia information retrieval systems retrieve only the relevant document, the result of this research will enhance and add additional feature to retrieve not only relevant documents but also the time frame in the document where the user search query is located.
- The speech corpus we prepared in achieving our research goal can help other researchers in doing the same or different research on Amharic ASR.
- The other benefit of this research is that it helps to view the effect of searching speech using query with the developed recognition error.
- Text-based STD can also help users to easily search a particular spoken term within a given audio file.
- The result of this research can help to develop applications to search and display the time interval where the spoken speech is located.

1.7. Scope and Limitation of the Study

The scope of this research is to show the performance of ASR on using manually and automatically segmented test speech and then developing text-based STD using bible and broadcast speech domains. Since we have a limited time and costly we used automatically segmented training Bible speech corpus for ASR performance comparison. The text-based STD system development was limited to showing and locating the time interval upon which the spoken word is located. Searching for a given user query word also doesn't include word sense disambiguation. On the other hand, the system was also limited to searching word using continuous speeches which didn't have any background music or nose. Searching the word query also does not include tasks like Amharic steaming rather perform exact matching of the query word.

1.8. Organization of the Thesis

The thesis is organized as five chapters including the presentation of the introduction section as Chapter-1. Chapter-2 presents literature review which includes the general overview that can help in understanding the problems and approaches. In Chapter-3, the methodology and clear steps that are followed to solve our problem statement are outlined. Chapter-4 covers experimental as well as result and discussion sections. Finally, conclusions and recommendations are presented in Chapter -5.

CHAPTER TWO

LITERATURE REVIEW

2.1. Amharic Language

Amharic is one of the Ethio-Semitic languages, which belongs to the Semitic branch of the Afro-Asiatic family that has the second large number of speakers in the world after Arabic. It is one of Ethiopia's most widely spoken Semitic languages, with at least 27 million native speakers [**19**]. The majority of the speakers of Amharic can be found in Ethiopia, but there are also a number of speakers in other nations, such as Israel, Eritrea, Canada, the USA and Sweden [**20**] [**21**].

2.1.1. Amharic Writing System

Unlike other sematic languages such as Arabic and Hebrew, Amharic is written from left to right. Present-day Amharic has acquired its composing framework from Ge'ez /gəʾəzə, which is still the classical and ministerial dialect of Ethiopia and uses a graphme based writing system called fidel /fidalə/ [**20**] [**21**] [**22**].

Amharic symbols are categorized into four different categories consisting 276 distinct symbols; these are core character, labiovelar, labialized and labiodental. Sample list of Amharic core characters are shown in Figure 2.1.

	Order						
	1st	2nd	3rd	4th	5th	6th	7th
	ə	u	i	a	e	i	o
h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
m	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
s	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
r	ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
s	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ

Table2. 1. Shows Sample Core Characters Used in Amharic Writing System with their Seven Orders.

2.1.2. Distinguishing Characteristics from other Languages (Amharic)

By nature, Amharic language has distinguishing characteristics from other languages which are spoken around the world. To begin with, Amharic has its own writing system which is based on phonetics. Also, Amharic language has its own characteristics of phonetics and phonological properties [**6**].

2.1.2.1. Amharic Phonology

The study of speech sounds used in various languages around the world is known as phonetics [**6**]. Amharic has thirty-one consonants which are generally classified as stops, fricatives, nasals, liquids, and semi-vowels.

Those sounds that are not found in English like ጵ are glottalized sounds. The existence of palatal consonants such as ሽ[S] and dental consonants like ት[t], lablized consonants which are pronounced by a slight round of the lips (ጕw), loan words ቨ[v] and the existence of geminated words make the language to have a distinctive characteristic from any other languages. The Amharic language has a total of 38 phones, including seven vowels and thirty-one consonants [**23**].

2.1.2.2. Consonants

Out of the 31 Amharic consonats, few of the Amharic consonants have similar phonetic transcriptions like English. These include ብ [b], ድ [d], ፍ [f], ግ [g], ህ [h], ክ [k], ል [l], ም [m], ን [n], ፕ [p], ር [r], ስ [s], ት [t], ቭ ው[w], [y] and ዝ [z]. They correspond to English consonants b,d,f,g,h,k,l,m,n,p,r,s,t,v,w,y,and z. In addition there are consonants that sound the same as English sounds but are represented using different symbols. These symbols includes ች [ch], ኝ [nx], ሽ [sx] and ዥ[zx]. Moreover there are also sounds which are the characterstics of Amharic but not found in English are ጵ [px],ጥ [tx],ፅ [xx],ጭ[cx] and ቅ [q] [**19**] [**13**].

Manner of articulation	**Voicing**	**Labials**		**Alveolar**		**Palatals**		**Velars**		**LabioVelar**		**Glottals**	
Stops	Voiceless	p	ፕ	T	ት			k	ክ	kwa	ኳ	Ax	ዕ
	Voiced	b	ብ	D	ድ			g	ግ	gwa	ጓ		
	Glottalized	px	ጵ	Tx	ጥ			q	ቅ	qwa	ቋ		
Fricatives	Voiceless	f	ፍ	S	ስ	sx	ሽ					H	ህ
	Voiced			z	z ዝ	zx	ዥ						
	Glottalized			xx	ጽ							Hwa	ኋ
Affricatives	Voiceless					c	ች						
	Voiced					j	ጅ						
	Glottalized					cx	ቼ						
Nasals	Voiced	m	ም	N	ን	nx	ኝ						
Liquids	Voiced			L	ል								
	Voiced			R	ር								
Glides		w	ው			y	ይ						

Table2. 2 Categories of Amharic Consonants **[**24**]**

2.1.2.3. Vowels

Amharic has a total of seven vowels, including five of the most common vowels (a, e, I o, and u), as well as two additional central vowels (E and I) shown in Table 2.2 [**6**] [**23**]. Vowels can be depicted in terms of the height of the tongue (high, mid and low), the horizontal position of the tongue (front, central and back) and the condition of the lips (rounded and unrounded) [**25**] [**26**].

	Front	**Central**	**Back**
High	ኢ[i]	እ [I]	ኡ [u]
Mid	-	ኤ [e]	ኦ [o]
Low	-	አ[a]	-

Table2. 3 Categories of Amharic Vowels

2.1.2.4. Amharic Morphology

Morphology is the study of word forms in terms of morphemes, which are the smallest semantic grammatical units [**27**]. The morphological phenomena of root patterns are used in Amharic. Here the root is a set of consonants and a pattern consists of a set of vowel inserted which are inserted among the consonants of the root [**6**] [**28**]. The Amharic languages words do have stem and affixes (prefix and suffix). Morphemes can be derivational or inflectional morphemes. Derivational morphemes can create new words in a language or they can change part of speech or lexical category from one to another. For the word teach+er, we can get teacher which is now by adding a new word to the verb teach we get a noun teacher. Inflectional morphemes are bound morphemes that serve a grammatical role in a language. Inflectional morphemes cannot create new words in a language or change the lexical category of a word in a language. The stem forms of the Amharic language can take many different forms [**6**] [**28**]. By adding suffix to the stem ሰብር it forms words like ሰበር-ኩ [I broke], ሰበር-ን[we broke], ሰበር- ሽ(feminine second person)[you broke], and the immediate object is identified as ሰበረ-ኝ[he brke me] as it is pointed by [**6**].

2.2. Audio Searching

Audio data implies all audible (being capable of heard) data like speech data, music, animal sounds, bell sounds, laughter, bird chirps, news footage archive, and audio lectures. Pitch is a generally slow-changing periodic signal in spoken speech that corresponds to the frequency of vibration of the vocal cords. Male pitch contribution is often between 50Hz and 250Hz, whereas female pitch contribution is typically between 120Hz and 500Hz [**29**].

On the audio file, searching will be done by giving query term. A query is usually a keyword or a short phrase that is given to the system for retrieving an audio file containing that query. Based on the query type, the searching techniques can be broadly classified into three: keyword spotting and spoken term detection and QBE STD as shown in Figure 2.1 [**30**]. Even if currently there are a lot of different tools, which help searching text, there are not for speech or audio search [**31**]. This requires to conduct researches on speech search that will enhance and ease the development of speech search/audio searches in general.

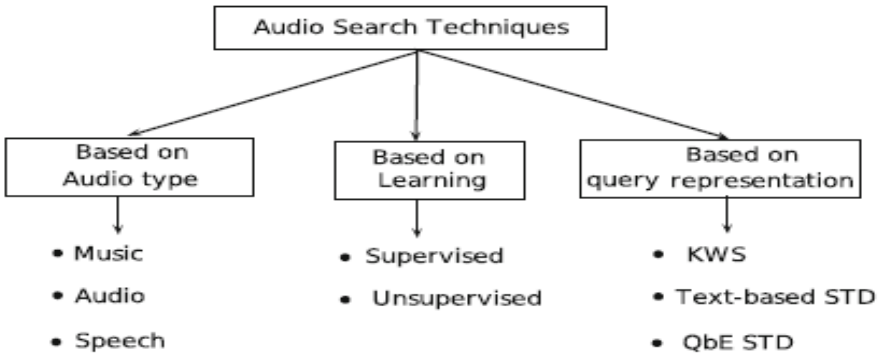

Figure 2. 1 Classification of audio search techniques **[**30**]**

2.2.1. Keyword Spotting

Keyword spotting (word spotting) allows finding the exact locations of a text query within a speech document or a speech stream [**3**].

In literature, sometimes the term keyword spotting has been used as STD. However, according to [**32**], in keyword spotting user query is known in indexing time whereas in text-based STD the query term is specified at search time. As a result, STD is more difficult to use since it has no prior knowledge of the queries that are being searched [**33**].

2.2.2. Text-based STD

The process of locating a particular search term from a collection of segmented speech is defined as STD. The general structure and components of STD systems are depicted in Figure 2.2.

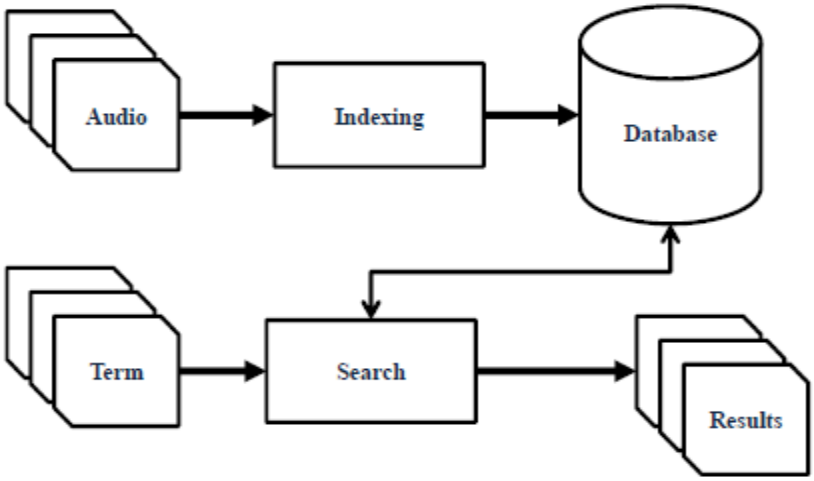

Figure 2. 2 General Structure of STD Systems **[**33**]**

From Figure 2.2., there are two steps to the STD procedure in general. The first stage, indexing, creates a database with an intermediate representation of the speech segments that are stored in a database. The search stage is then in charge of finding putative (acceptable) occurrences of query words in this intermediate database. The search should be carried out quickly and precisely [**33**]. Spoken Term Detection (STD) has advantages like it offers the possibility of retrieving any speech file that contains any term (a sequence of one or more words) from its textual representation, allowing the search of any term in a large index efficiently. This technology can be accessed using any device with text input capabilities [**34**] [**10**].

2.2.3. Query-By-Example STD

A user presents the system with desired audio snippets containing queries. The system then searches the database for segments that closely resemble the query [**30**]. Table 2.4 depicts summery of different audio searching techniques with the query type to use and whether it needs speech to text conversion using ASR.

Method	Query type	Need for speech to text conversion
KWS	Predefined word/phrase	Yes
Text-based STD	Unspecific word/phrase	Yes
QbE-STD	Unspecific word/phrase	No

Table2. 4 Summary of Audio search techniques

2.3. Challenges on Spoken Term Detection

The state-of-the-art STD systems are usually based on a Large Vocabulary Continuous Speech Recognition (LVCSR) engine and search for keywords in the results returned by the engine. The search for Out-of-vocabulary (OOV) words remains a challenging problem since the OOV words are always misrecognized by the LVCSR engine [**35**]. OOV refers to words that are not in the lexicon and is the most common source of error in ASR [**20**] [**36**].

Different approaches are there to minimize the OOV effect on the ASR. One way of achieving high lexical coverage is building LM (Language Model) on morpheme level [**37**]. The OOV rate obtained on the Amharic language which was done by [**20**] on word-based and morpheme-based recognition is 28.18% and 6.28% respectively. So implementing and using morpheme-based morfesor will help in reducing the OOV rate that will be caused by the result of ASR. The other way of minimizing OOV is applying and using close vocabulary.

2.4. Automatic Speech Recognition

Command and control, dictation, transcription of recorded speech, searching audio documents, and interactive spoken conversations are just a few of the uses for automatic continuous speech recognition (CSR). A collection of statistical models describing the various sounds of the language to be identified is at the heart of all speech recognition systems.

Since speech has a temporal structure and can be encoded as a sequence of spectral vectors spanning the audio frequency range, the hidden Markov model (HMM) provides a natural framework for constructing such models [**38**] [**39**].

2.4.1. Category of Speech Recognition

Automatic speech recognition is one of the most automatic speech processing areas, allowing the machine to understand the user's speech and convert it into a series of words through a computer program, thus creating a kind of natural communication between man and machine [**40**].

2.4.2. Speaker Dependent / Speaker Independent

Speech recognition systems are classified into two categories, speaker dependent and speaker independent [**2**].

Speaker dependent systems are trained by the individual who will be using the system. These systems are capable of achieving a high command count and better than 95% accuracy for word recognition. The drawback to this approach is that the system only responds accurately only to the individual who trained the system. This is the most common approach employed in software for personal computers.

Speaker independent is a system trained to respond to a word regardless of who speaks. Therefore the system must respond to a large variety of speech patterns, inflections and enunciation's of the target word. The command word count is usually lower than the speaker dependent however high accuracy can still be maintain within processing limits. Industrial requirements more often need speaker independent voice systems, such as the AT&T system used in the telephone systems.

2.4.3. Recognition Style

Speech recognition systems have another constraint concerning the style of speech they can recognize. They are three styles of speech: isolated, connected and continuous and spontaneous.

Isolated speech recognition systems can just handle words that are spoken separately. This is the most common speech recognition system available today. The user must pause between each word and command spoken.

Connected is a halfway point between isolated words and continuous speech recognition. **Continuous** is the natural conversational speech we are used to in everyday life which is pronounced naturally [**17**]. It is extremely difficult for a recognizer to shift through the text as the word tends to merge. For instance, "Hi, how are you doing?" sounds like "Hi,.howyadoin".

In summary types of Speech in most studies, speeches are resumed into four types: [**41**] [**12**]

• Isolated Words: This type usually requires a quiet (silence state) between utterances.

• Connected Words: Word systems are similar to isolated words, the only difference between themes is to allow separate words to "run together" with a minimum of pausing between them.

• Continuous Speech: The users of this type talk almost normally, while the computer selects the content. It is one of the most difficult systems.

• Spontaneous Speech: At a basic level, it can be thought of as speech that is natural-sounding and not rehearsed.

The size and volume of the Vocabulary used in the speech recognition system are important because it affects the complexity and processing requirements and determines the accuracy of the ASR system. To simply define [**42**]:

• 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 tens of thousands of words.

2.4.4. Speech Recognition Architecture

The speech recognition system includes components like acoustic front-end, acoustic model, lexicon, language model and decoder as depicted in the figure below. The acoustic front-end convert's speech signal into appropriate features used by the recognizer. The process of converting the audio wave form into a sequence of fixed-size acoustic vectors is a process called feature extraction. Feature vectors are generally generated every 10 milliseconds using a 25 millisecond overlapping analysis frame. 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 as an acoustic model P (O|W) and P (W) is determined by a language model [**38**]. The process of establishing statistical representation for the feature vector sequences is computed from the speech waveform. Hidden Markov Model (HMM) is one of the most commonly used statistical models to build acoustic models [**38**].

2.5. Language Modeling

Natural language contains a large number of words and terms, which can lead to a variety of ambiguities. There are several types of ambiguity like lexical ambiguity, syntactic ambiguity and anaphoric ambiguity. Natural languages are completely comprehended by humans,

notwithstanding their uncertainties. Machines, on the other hand, are incapable of processing the ambiguities in real human language. As a result, language models are employed to convert text into a machine-readable format [**43**]. Modeling has always relied on two main approaches. The first is based on grammar, such as context-free or unification grammars, which are defined using linguistic knowledge. The second method employs probabilistic models based on corpora, which have been widely used in natural language processing since their inception in the 1980s [**44**]. They're mostly used to determine how frequently a sequence of tokens in a corpus, which is a collection of texts, such as a document, occurs [**45**]. To put it another way, a probabilistic model is created to aid in the prediction of the next word from a given sequence of words. Let's take a look at a statement that is just partially completed. Please submit your application. It's more likely that the following word will be homework or a paper, rather than Professor [**43**].

2.5.1. Smoothing Techniques

The word "smoothing" refers to a set of procedures for fine-tuning the MLE (Maximum Likelihood Estimation -that is by counting events in context on some training corpus) to produce more accurate probabilities. Some of the smoothing techniques are (Laplace Smoothing, Add λ Smoothing, Natural Discounting, Good-Turing Smoothing, Interpolation and Backoff) which solve the problem of data sparsity based on the raw frequency of n-grams. Details of each smoothing technique are clearly elaborated in [**44**].

2.5.2. Interpolation

Interpolation is a smoothing technique used to solve problem of data sparsity using n-gram hierarchy [**44**]. M. Y.Tachbelie [**44**] combines the probability estimates of all n-gram orders based on the assumption that if there isn't enough data to estimate a probability in the higher-order n-gram, the lower-order n-gram can frequently give relevant information. In simple linear interpolation, they estimated the tri-gram probability p (wn|wn−1wn−2) that, by mixing together the uni-gram, bi-gram, and trigram probabilities, each weighted by a λ shown with equation 2.1 as:

$$P^*(w_n|w_{n-1}w_{n-2}) = \lambda_1 p(w_n|w_{n-1}w_{n-2}) + \lambda_2 p(w_n|w_{n-1}) + \lambda_3 p(w_n) \quad (2.1)$$

Where

$$\sum_i \lambda_i = 1 \quad (2.2)$$

Accordingly, lambda value was calculated based on the cotext/history.it was also elaborated with example that, if the context of a particular tri-gram is frequently observed, then a high λ will be appropriate for the tri-gram, and the tri-gram was given more weight in the interpolation. On the other hand, for a single occurrence of history, a lower λ will be suitable [**46**] [**44**].

2.6. CMU Sphinx

CMU Sphinx is an open source speech recognition toolkit developed by Carnegie Mellon University's Sphinx group, Sun Microsystems Laboratories, Mitsubishi Electric Research Lab (MERL), and Hewlett Packard (HP), with contributions from the University of California at Santa Cruz (UCSC) and Massachusetts Institute of Technology (MIT). CMU Sphinx created several different versions. The Sphinx4 was developed in 2005 voice recognition library which is written entirely in Java. With the aid of CMU Sphinx acoustic models, it provides a simple and fast API for converting voice recordings into text. It may be utilized on servers as well as in desktop software. Sphinx4 aids in the identification of speakers, the adaptation of models, the alignment of existing transcription to audio for time stamping, and many other tasks. Sphinx 4 uses models trained by Sphinx 3 trainer and also recognizes isolated and continuous speech
[**17**] [**47**]. The other versions Sphinx 1, Sphinx 2(high-speed large vocabulary speech recognizer), Sphinx3 (slower than Sphinx 2, but provides more accurate Large Vocabulary Speech Recognition System), PoketSphinx (fastest version of CMU Sphinx speech recognition system that uses semi-continues output PDFs with HMM which can be used in devices and live applications and it is as accurate as Sphinx 3 and Sphinx 4 [**47**]. However, the performance of Sphinx 4 is less when compared with Sphinx 3 [**48**].

2.7. Speech segmentation (pydub)

Python’s pydub is a speech segmentation package based on silence detection on the energy. By defining a period of silence as the time duration when the Root Mean Square (RMS) power of the speech signal drops below a given dB value such as -35 for at least a value of some seconds like 500ms silence of a given speech would be detected [**16**] Apart from the automatic speech segmentation, manual speech segmentation could be done using Auddacity software. This software is freely available open-source digital audio editor and recording application software.

2.8. Performance Evaluation

Evaluation of speech recognition is important for any research to check its respective performance. Speech recognition could be measured using speed and accuracy. The speed of ASR could be measured using the Real-time factor (RTF). On the other hand accuracy of the recognition system can be measured using WER (Word Error Rate) and WRR (Word Recognition Rate).WER is most the widely used metric [**41**] [**49**] [**50**].This is largely affected by non-linguistic conditions, such as noisy environment, variety of recording environments, sound effects and multiple speakers [**51**].

$$\text{Word Error Rate}(\%) = \frac{\text{Insertion(I)} + \text{Substitution (S)} + \text{Deletion (D)}}{\text{No. of Reference Words (N)}} * 100 \quad (2.3)$$

From equation 2.3, S refers is the number of substitutions performed in the output text as compared to the ground truth. D refers the number of deletions performed, and I is the number of insertions performed. N is the total number of words in the ground truth. The other performance metrics WRR could be calculated using the formula which is stated in equation 2.4.

$$\text{Word Recognition Rate (WRR)}=1 - \text{WER} = \frac{N - S - D - I}{N} \quad (2.4)$$

2.9. Related Previous Studies

In this section, previous studies related to the present thesis work are highlighted. Specifically, publications related to searching for specific utterance using written text as query word and speech segmentation on ASR and search on speech were explored and summarized.

2.9.1. Speech Segmentation

J. Neto et.al [**52**] developed ASR system for automatic speech transcription applied to a Broadcast News (BN) task for the Portuguese language. The researchers developed ASR using prepared speech with low background noise and good quality audio. The results also included WER obtained in all test sentences; including noise, music, spontaneous speech, telephone speech, nonnative accents and F0 focus condition sentences. In order to develop the system the researchers followed a hybrid approach which combines use of HMM and MLP(Multi-Layer Perceptron).After the ASR development, the researchers further checked the performance of the system using automatically and manually segmented speech, since automatically segmented speeches are not perfect. A 29 minute

test set speech was used and compares three transcription results. First 241 manually transcribed sentences, second was done by considering the whole program as one sentence where no preprocessing was made, and the last test set is segmented automatically and produce 366 sentences. The researchers recorded WER of 26.9, 27.1, and 29.0 respectively.

C.Liu et.al [**49**] compares the relative performance of ASR systems developed using automatically and manually transcribed speech corpus. The researchers use two sets of manual transcriptions and five sets of automatic transcriptions (Google Cloud, IBM Watson, Microsoft Azure, Trint, and YouTube) for comparison aimed at helping other researchers or research community to select for accurate transcription services. The researchers use two Simulated Patient (SP) one male and one female. This trained SP is regularly interviewed by student doctors. They use the 12 randomly selected where gender is put into consideration from a total of 84 conducted interviews. Then the researchers transcribe manually using independent professional transcribers and hand-picked freelancers available at Rev.com (Rev). From the two transcription results, better transcription is selected by comparing the WER which was done using open source asr-evaluation library. The researchers obtain result which shows manual transcription is better than all other transcriptions (Google Cloud, IBM Watson, Microsoft Azure, Trint, and YouTube) even if youtube offers accurate transcription compared with others.

Nitza Geri et.al [**51**] investigate how to significantly reduce the gap between machine and human performance for Hebrew text navigation through search terms. The purpose of their study was to examine rapid and affordable ways to transcribe Hebrew speech, by existing tools, and to explore their potential to provide good enough, not perfect though video transcriptions. In solving their stated hypothesis they used the already available state-of-the-art speech recognition models Google/HTML5 speech recognition system for Hebrew and Nuance Mobile Developer Program – NDEV. A total of 40 minutes of Hebrew speech was used for their ASR experiment by using the above ASR engines. From their first experiment, they found that the WRR tests showed that the ASRs performed better with read speech than with lectures, in quantity and quality.

R.Mekonen [**13**] has developed a sentence-level automatic speech segmentation system for Amharic which is used to segment the spoken speech into sentence level. To implement the sentence level segmentation two approaches were used by the researcher. In the first approach, the researcher used

an automatic tool for segmenting and labeling Amharic speech data. An acoustic model is created using speech and their text scripts and compiling them into a statistical representation of sounds that make up words. To implement the sentence level automatic segmentation system the researcher uses 4 hours of speech which are collected from different domains and speech types. Amharic bible, broadcast news, broadcast conversation and Amharic fictions are different domains where the corpus was collected and spontaneous and read speech are the type of speech's used/collected by the researcher. In order to implement the sentence level, automatic segmentation system the researcher uses two approaches. On the first approach, preprocessing rule-based segmentation using Audacity software which is used to segment an audio file and given to the different acoustic models. These acoustic models are monosyllable, Tied-State syllable and monophone acoustic models. Then the audio file with its respective transcribed file is given to the Forced aligner and the segmentation result is displayed. In second approach the researcher uses features such energy and F0 features are combined with seven prosodic features (rate-of-speech, volume change rate, pause, succeeding and preceding sentence duration, succeeding and preceding pause duration, and rate of-speech duration) to detect sentence boundaries. Then adaBoost algorithms are used to check the performance and accuracy of the supervised classifier. Following the two different approaches, the researcher found results of the first approach that is the rule-based approach are better than the second approach which shows a better accuracy.

A.Rajpoot and P. Sharma [**53**] do research to segment the speech signal into silence, voiced and unvoiced regions which are aimed at increasing the performance of recognition systems. To achieve their research objective they propose an algorithm that is fast and simple. They developed an algorithm using various speech features such as Zero Crossing Rate (ZCR) which is the number of times the amplitude of a speech signal passes through a value of zero in a given time interval or frame, Short Time Energy (STE) which helps to quantify how much energy is in a speech signal at any given moment here high for voiced and low for unvoiced and 0 for silent [**54**], and Fundamental Frequency (F0) which is the quality of pitch that rises and then falls as something is spoken voiced. The algorithm they applied to 15 selected Hindi words spoken by four persons (3 male and 1 female) and each spoken 3 times. The researchers use MATLAB 2011a to implement the algorithm they developed and reached an accuracy of 96.61 %. The algorithm's accuracy was determined by comparing the number of samples correctly defined in the spoken word to the manual classification of the voiced, unvoiced, and silence regions in the word, and then dividing that number by the total

number of samples. According to the researchers, the remaining error was due to little noise or lower energy during the starting and ending of the word.

C.V. Heerden et al. [**36**] Experiment on sub-word unit syllable-like and morpheme-like units for different languages. The aim of their research was on how to reduce out-of-vocabulary (OOV) keywords which are generated by the ASR. The researchers compare Syllable-based units and Morpheme-based units(two approaches) for Spoken term detection for OOV for Amharic ,Guarani(official languages of Paraguay), Igbo(spoken in southeastern Nigeria), Javanese(spoken in Malaysia, the Netherlands, and Singapore),Dholuo (spoken in southwestern Kenya), Mongolian (spoken in Mongolia a),Pashto(Eastern Iranian) languages. The corpora used in their experiments were the "Full language packs," which were distributed in the fourth year of the IARPA (Intelligence Advanced Research Projects Activity) project BABEL and each contained about 40 hours of training data for all languages and Kaldi was used as a speech recognition toolkit. From the result, they obtain when comparing OOV results, whether to use syllables or morphemes becomes a language-specific choice. For Amharic, Dholuo and Pashto, the morpheme-based OOV results are best, while for the rest of the languages, syllable-based OOV results are slightly better (Guarani, Igbo, Javanese) to significantly better.

2.9.2. Searching on Speech File Using Text as Query Word

In this part, we try to explore related research works done by different researchers which are related to a part of our research work, which is speech search using text query. We also summarize researches that are related with our research.

K. Kodlekere et.al [**7**] present an interactive media player that enables the user to perform offline audio content based searching capabilities with a given multimedia file. As for the researchers, one reason for doing their research was to alleviate the problem of Massive Online Courses (MOOCs) which is a free online course available for anyone to enroll. However, out of the massive registered users, only 15 % are complete their course. To improve the performance of completion of the course researchers come up with a novel approach of making a multimedia file viewer-friendly (multimedia player) which decreases consumption time. Audio files do have metadata that shows the title, file type and size. All this metadata doesn't tell what content the audio file has. So for the users to check for the content of the multimedia file users have to navigate to the entire file which is a time-consuming process. To solve searching content by metadata (title) researchers propose a system consists of a media player which assists the user to search for a keyword within the video, which also

helps the user navigate within the video and browse for a topic of their interest. The proposed system works by accepting the multimedia file and input keyword. Then the coming audio file from a user is converted into a format that will be useable by the system i.e.wav form and passed to the ASR.ASR transcribes the coming multimedia file and indexing and searching for the keyword, which will return the keyword with its respective time frame. By using the time frame which was displayed on the developed media player (using python) users can further search audio content to locate the specific utterance. For the development of their proposed system researchers uses different tools. For ASR development they use sphinx over IBM's(International Business Machines Corporation) Watson and Google's speech recognition API (Application Programming Interface), a software interface that allows two different applications to interact or communicate with each other. Researchers use sphinx because it is open-source and easily accessible to anyone. To develop the Media player they used LibVLC API which enables multimedia capabilities by embedding it to the application. For the user interface development, they used gtk+ which is a python library. From the result obtained from their research words with a confidence level above the set threshold were considered as true hits and those with a confidence level below it are considered as false hits. The accuracy of the keyword spotter is dependent on the confidence value chosen as the threshold for each word which is 0.85. The proposed system identifies keywords on average 60-65%. As of the researchers, this result can be improved by using or selecting high-quality videos.

A. Hassen [**8**] Propose an Amharic speech search engine by designing an Amharic speech document. Since most search engines are designed for English and they struggle to find documents written in Amharic. The prototype developed by the researcher has four basic components such as crawler, audio processing, indexer and query engine. The first component identified by the researcher is crawler which is responsible for crawling through the web pages and downloading speech documents from the web. JSpider were used by the researcher to download speech documents from the web since it is an open-source and java configurable tool. Websites of Sheger FM, Bisrat 101.1 and Voice of America (VOA) Amharic were a seed URL used by the researcher where web crawler will begin to traverse a site. Speech Content Processing is the second component, which is responsible for identifying Amharic speech files, indexing the coming Amharic multimedia files and searching the indexed document using the query text through the developed user interface. On Speech identification the process in which Amharic web audio documents are filtered where Amharic contents are greater than 60% i.e. if the audio documents Amharic content is less than 60% it will not be selected and left out from further processing. To filter the Amharic audio document the researcher

uses a Tika content extractor. Once the required document is filtered transcription of the Amharic audio content into its respected Amharic text and saved in the transcription file. To do that the researcher uses the already developed model. The speech recognition model was developed using sphinx 4 which is an open-source tool for recognition. The speech recognition model was responsible for speech transcription. The other task which is done on the speech processing part is indexing. It is a method of indexing documents in a repository into an effective cross-reference lookup. The indexing was developed by solr indexer which uses an inverted index. The third component of the Amharic speech search engine is a Query Engine. This component is responsible for matching and ranking the transcribed document. Query processor is the user interface part where it accepts Text-based Speech Query and it will be matched to the indexer. Preprocessing (tokenization stop word removal and others) is done on text-based speech query to increase to get a better search result. The experimental results revealed that the Amharic speech retrieval engine had an accuracy of 80% on the top ten results and a recall of 92% as compared to its corresponding retrieval engine.

I.Ayass et al. [**55**] Propose a system that can automatically process the YouTube video file to identify the discussed topics and allows users to access that information in a significantly shorter time through a search engine. The topic of interest is described by keywords that are used for searching the video or audio material. The proposed system works in a way that first cheek the coming multimedia data. If it is a video file and in *.wav format it will automatically be transcribed to its respective English text. From the transcribed text respective topic will be deduced. After transcription matching the indexed topic with the predefined list of topics (keyword spotting) was made. Then matching results from the previously mentioned process and the respective video file will be stored in the database, which will be used by the users while requesting the system by entering text query word. CMU Sphinx was used to transcribe the video into its respective text file. Researchers select this ASR toolkit because it is open-source and easy to integrate. The transcription accuracy of the proposed system was 50-80%.

A. Mohammed [**9**]developed a simplified search engine that enables search operation that will search the audio and video file content which was usually done using file title and description. While developing the engine, first the audio and video file was converted into a time-marked stream of text and then searching is made. Here the search engine has three parts. The web archiving indexers part, which is similar to a website with the ability to play the media from its

source without storing or copying the file. The second part is the speech to text recognizer engine which is responsible to recognize the played video & audio played before by the indexers and returning the recognized text, which is stored in the Database. The third part is the user interface and matching engine which is responsible for taking the user search query and then machining it with the text fields in the database, which is, quite similar to text-based search engines.

The other research made by [**56**] was the Turkish Broadcast News transcription and retrieval system. On the development Out-of-Vocabulary (OOV) was a challenge where even sub-word-based recognition units are utilized. To alleviate this problem and to increase the accuracy the researchers use moderate size vocabularies according to the researchers which even performs better than a vocabulary size of 500k. The researchers developed a Spoken Term Detection system and a Spoken Document Retrieval system. To retrieve the spoken data ASR was used.

Summery

In all the above related works, to the knowledge of the researchers, there is no research conducted on searching on a speech on a given Amharic audio file, even if there are researches conducted for other languages like English [**7**]. In their research keyword spotting were used as a searching technique where the speech recognition was trained with most frequently used words or keyword. The coming audios are directly given to the ASR where segmentation of a speech is not considered in their research. This research has a twofold benefit. First, it enables us to check the performance of ASR by giving segmented audio files to the ASR. Second, it allows searching and displaying the time interval in which the spoken speech is located.

So from the above review, we try to include segmentation in order to verify whether automatic segmentation has an impact on ASR and on Amharic audio speech files since music files do have a negative impact on the accuracy of the ASR.

Speech segmentation			
Title	**Year**	**Remark**	**Researchers**
Automatic Speech Recognition Systems and the Nonverbal Responses to Unintelligible Speech	2019	Compares the performance of ASR using simulated patents (SP) interview record and by using manually transcribed SP records as a reference.	C.Liu et.al [**49**]
Can automatic speech recognition be satisficing for audio/video search? Keyword-focused analysis of Hebrew automatic and manual transcription	2014	There pourpose of study was to examine rapid and affordable ways to transcribe Hebrew speech	Nitza Geri et.al [**51**]
Automatic identification of silence, unvoiced and voiced chunks in speech	2013	Segmentation	A.Rajpoot and P. Sharma [**53**]
Automatic Speech Annotation and Transcription in a Broadcast News task	2003	Development of ASR along with comparing the automatically segmented speech with the manually segmented speech.	J. Neto et.al [**52**]
Searching			
Develop an Audio Search Engine for Amharic Speech web Resources	2019	Search engine	A. Hassen [**8**]
Prosody Based Automatic Speech Segmentation for Amharic	2019	Speech segmentation	R.Mekonen [**13**]
Keyword Based Indexing of a Multimedia File	2017	Search a speech using text keyword and returns time frame and segment of video	K.Kodlekere et.al [**7**]
Constructing sub-word units for spoken term detection	2017	Compares Syllable-based units and Morpheme-based units for Spoken term detection for OOV	C.V. Heerden et al. [**36**]

Audio Indexing for YouTube	2015	Text based searching system about discussed topic in audio file	I.Ayass et al. [**55**]
Video & Audio Content Search Engine (VACSE)	2014	Audio search engine	A.Mohammed [**9**]
Turkish Broadcast News Transcription and Retrieval	2009	Audio retrieval	M. Saraclar et.al [**56**]

Table2. 5 Summary of related works

CHAPTER THREE

METHODOLOGY

In this section, the procedure we followed to do a speech search using Amharic query text is discussed. Searching does not only include or display the user's query text but also displays the time frame where spoken speech is located. We followed experimental research methodology to conduct our study. The first experiment was done by checking the performance of automatically and manually segmented test speech on ASR. This can give a clear understanding for the implementation of text-based STD and what effect we will face up on using automatically segmented speech for text-based STD development.

3.1. General Workflow

The general approaches and procedures we followed towards reaching research objectives are depicted in Figure 3.1.

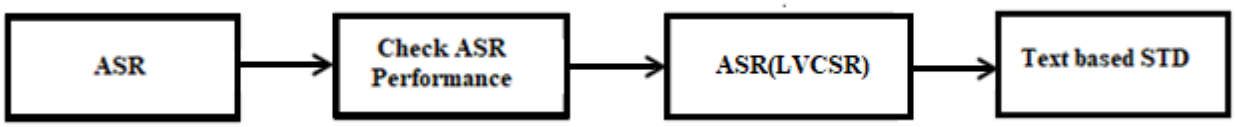

Figure 3. 1 Block diagram for the General workflow of text-based STD development

As shown in Figure 3.1., we followed four steps to achieve our objective. The first step was performed to develop ASR using Bible speech corpus. Then, the second step was for checking the performance of the developed ASR for selecting the optimal automatic speech segmentation. In the context of our work, optimal segmentations is defined as the lowest WER obtained after performing comparisons between manually and automatically segmented test speech.

The performance of ASR was checked using automatically and manually segmented bible speech corpus (test speech corpus). Once the ASR performance was established from the result, the third step was followed to develop a model using LVCSR [**14**]. The final step was necessary since we used the acoustic models of the ASR for text-based STD development. Details of each step are presented in the following sections.

3.1.1. Corpus Preparation

This section presents discussion on the speech corpus we used and that we prepared for the comparison of ASR performance up on using automatically and manually segmented bible speech corpus.

The block diagram is shown in Figure 3.2., depicts training and test set speech corpus used and prepared for ASR performance comparison. It also shows how the decoder transcribes a given test speech sets using acoustic model, language model and the phonetic dictionary which are components of ASR [**57**]. Both the training and test sets (corpus) were prepared from Bible domain. For the test speech, we used similar speeches (in content) except we segmented them manually and automatically. Finally, we checked the performance of ASR upon using the manually and automatically segmented speeches. Details of the corpus preparations are explained in the subsection of this section.

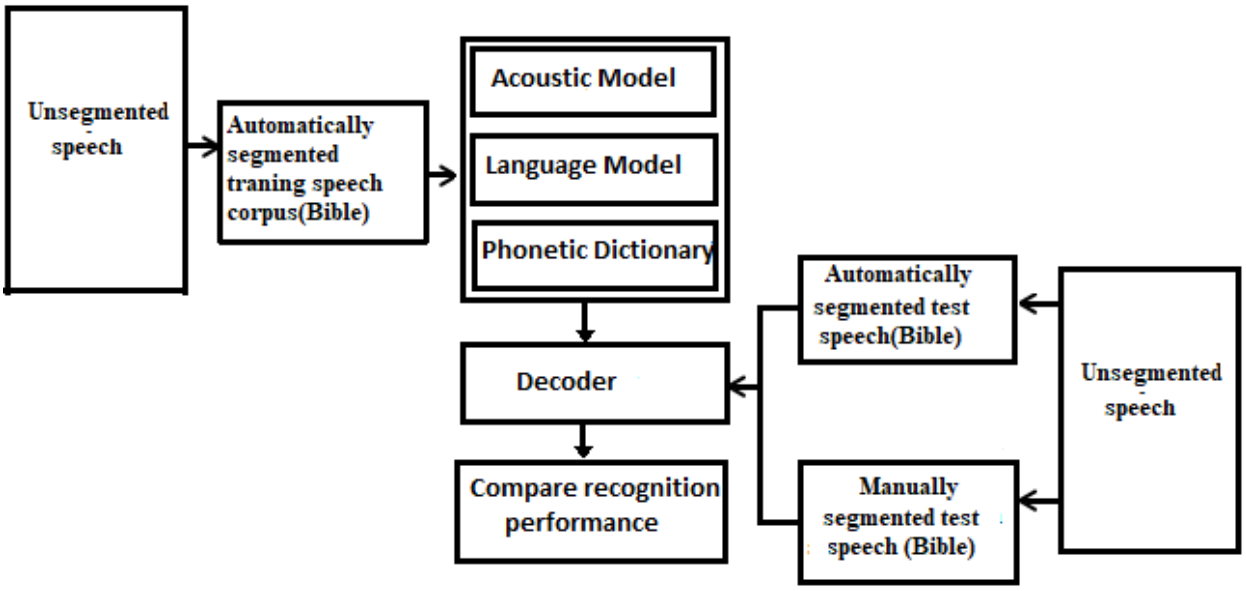

Figure 3. 2 General flow of automatically and manually segmented speech corpus preparations to check the ASR performance.

3.1.1.1. Training Speech Corpus

In developing the ASR model, we used a 1 hour and 35-minute audio file. Then, we segmented, copied and pasted a total of 1050 sentences. This file was downloaded from a publicly available YouTube video [**58**]. To prepare training transcription, we first automatically segmented a 1 hour

and 35-minute file using python pydub with minimum silence of 600 and silence threshold of -35 and transcribed. The transcription was made by listening to the starting and ending of the segmented audio file and matching it with the respective text which was available at WordProject [**59**] . The text has been manually segmented to be aligned with the segmented speech. Sample training speech transcriptions are shown in appendix B of the report.

3.1.1.2. Test Speech Corpus

We prepared two different test speech corpuses: manually segmented speech and automatically segmented speech. The reason for the two different test speech corpus was to check the performance of ASR using the same speech file. A total of 30-minute bible speech was used for both manual and automatic speech segmentation. Sample test speech transcriptions are shown in Appendix C of the report.

3.1.1.2.1. Manual Speech Corpus Preparation

The manually segmented speeches were used from the research done by [**13**] . Even if we found lots of manually segmented speech from the indicated source, we only used the first 30- minute of the segmented speech. The speech was segmented into small pieces by the author through reading and checking where the sentence begins and ends by listening to the speech. Segmentation was made by audacity software by the author. We took 307 segmented files, which were obtained when the 30- minute unsegmented speech was manually segmented with a sample rate of 22050 Hz.

Since the research done by Mekonen [**13**] used HTK as a development tool, the format and files couldn't be used for our ASR development using Sphinx. Thus, the speech was made in a format to be used by sphinx speech recognition tool. Therefore we manually segmented the text to be aligned with the segmented speech. We used the bible texts from WordProject [**59**]. Table3.1. Shows transcribed text, the audio file and duration of every automatically segmented audio which will further be used by the ASR tool. Apart from transcription, for this particular task, further preprocessing was made like converting the sample rate from 22050 Hz to 16000 since our configuration for sphinx was based on 16000 sample rate.

No	Manually segmented file	Copy pasted text	Duration
1	mat00.wav	ኢየሱስም በይሁዳ ቤተ ልሄም በንጉሱ በሄሮድስ ዘመን በተወለደ ጊዜ እነሆ ሰብአ ሰገል የተወለደው የአይሁድ ንጉስ ወዴት ነው	00:08
2	mat01.wav	ኮከቡን በምስራቅ አይተን ልንሰግድለት መጥተናልና እያሉ ከምስ ራቅ ወደ ኢየሩሳሌም መጡ	00:06
3	mat02wav	ንጉሱ ሄሮድስም ሰምቶ ደነገጠ	00:03
4	mat03.wav	ኢየሩሳሌምም ሁሉ ከእርሱ ጋር	00:03
5	mat04.wav	የካህናትንም አለቆች የህዝቡንም ጻፎች ሁሉ ሰብስቦ ክርስቶስ ወዴት እንዲወለድ ጠየቃቸው	00:07

Table 3. 1 Sample test transcriptions using a manually segmented audio file

3.1.1.2.2. Automatic I (a sentence like segmentation)

The automatic segmentation of unsegmented speech is done using silence. This was done using the pythons pydub package. A 30-minute unsegmented speech was segmented automatically using parameters of minimum silence of 600 and silence threshold of -35. The automatic segmentation was done using the pythons pydub package. The library segments the audio files according to the minimum silence and threshold parameters. Once we segmented automatically, the next step was transcribing the automatically segmented speech. In our case, since we have an online source of respected text for the segmented speech, we simply made the transcription by listening to the starting and ending of the manually segmented speech. Then, we copied and pasted the corresponding text from WordProject [**59**]. The transcription also resulted in 369 sentences. Sample transcription for segmented *.wav files are depicted in Table3.2.

No	Automatically segmented file	Copy pasted text	Duration
1	MatC21.wav	ኢየሱስም በይሁዳ ቤተ ልሄም በንጉሱ በሄሮድስ ዘመን በተወለደ ጊዜ	00:04
2	MatC22.wav	እነሆ ሰብአ ሰገል የተወለደው የአይሁድ ንጉስ ወዴት ነው	00:03
3	MatC23.wav	ኮከቡን በምስራቅ አይተን ልንሰግድለት መጥተናልና እያሉ ከምስ ራቅ ወደ ኢየሩሳሌም መጡ ንጉሱ ሄሮድስም ሰምቶ ደነገጠ	00:08
4	MatC24.wav	ኢየሩሳሌምም ሁሉ ከእርሱ ጋር የካህናትንም አለቆች የሕዝቡንም ጻፎች ሁሉ ሰብስቦ	00:07
5	MatC24.wav	ክርስቶስ ወዴት እንዲወለድ ጠየቃቸው እነርሱም አንቺ ቤተ ልሄም የይሁዳ ምድር	00:06

Table 3. 2 Sample test transcriptions using automatically segmented audio file

3.1.1.2.3. Automatic II (word /phrase like segmentation)

The automatic segmentation of unsegmented speech was done using silence. This was performed using the pythons pydub package. A 30-minute unsegmented speech is segmented automatically using parameters of minimum silence of 400 and silence threshold of -26. A total of 877 segmented *.wav files were obtained by passing the parameters of minimum silence and threshold parameters to the functions inside pydub. These 877 results were mostly phrase and word-level segmentation. The procedure that was used for sentence-like segmentation was repeated for the phrase-like segmentation. Sample transcription for the segmented *.wav files are depicted in Table3.**3.**

No	Automatically segmented file	Copy pasted text	Duration
1	Sseg1.wav	ኢየሱስም በይሁዳ ቤተ ልሄም በንጉሱ በሄሮድስ ዘመን በተወለደ ጊዜ	00:04
2	Sseg2.wav	እነሆሰብአ ሰገል	00:01
3	Sseg3.wav	የተወለደው የአይሁድ ንጉሥ ወዴት ነው	00:01
4	Sseg4.wav	ኮከቡን በምሥራቅ አይተን ልንሰግድለት መጥተናልና እያሉ	00:03
5	Sseg5.wav	ከምሥራቅ ወደ ኢየሩሳሌም መጡ	00:01

Table 3. 3 Sample test transcriptions for automatically segmented audio files (a phrase like/word-like)

3.1.1.2.4. Merging the Automatically and Manually Segmented Speeches

Here we try to show how it looks after processing the manually and automatically segmented speeches. Table 3.4 shows the starting and ending word of a segment for both the manually and automatically segmented speeches. This segmentation was obtained by changing parameters of the minimum silence and silence thresholds. We first tried to segment speech in word-like and phrase-like segments. However, through changing the parameters, we later tried to segment speech with minimum silence of 600 and silence threshold of -35 which gives sentence-like segments which is used in other part of our research. The comparison for the Automatic I and Automatic II could be seen in Table 3.4 and Table 3.5 respectively.

No	Automatic I			Manually segmented speech		
	Wav file	**Duration**	**Transcribed(Copy pasted) text**	**Transcribed(Copy pasted) text**	**Duration**	**Wav file**
1	MatC21.wav	00:04	ኢየሱስም በይሁዳ ቤተ ልሄም በንጉሱ በሄሮድስ ዘመን በተወለደ ጊዜ	ኢየሱስም በይሁዳ ቤተ ልሄም በንጉሱ በሄሮድስ ዘመን በተወለደ ጊዜ እነሆ ሰብአ ሰገል የተወለደው የአይሁድ ንጉስ ወዴት ነው	00:08	mat00.wav
2	MatC22.wav	00:03	እነሆሰብአ ሰገል	ኮከቡን በምስራቅ አይተን ልንሰግድለት መጥተናልና እያሉ ከምስ ራቅ ወደ ኢየሩሳሌም መጡ	00:06	mat01.wav
3	MatC23.wav	00:08	የተወለደው የአይሁድ ንጉሥ ወዴት ነው	ንጉሱ ሄሮድስም ሰምቶ ደነገጠ	00:03	mat02wav
4	MatC24.wav	00:07	ኮከቡን በምሥራቅ አይተን ልንሰግድለት መጥተናልና እያሉ	ኢየሩሳሌምም ሁሉ ከእርሱ ጋር	00:03	mat03.wav
5	MatC24.wav	00:06	ከምሥራቅ ወደ ኢየሩሳሌም መጡ	የካህናትንም አለቆች የህዝቡንም ጻፎች ሁሉ ሰብስቦ ክርስቶስ ወዴት እንዲወለድ ጠየቃቸው	00:07	mat04.wav

Table 3. 4 Manually and automatically (automatic I) segmented speeches in parallel

No	Automatic II			Manually Segmented speech		
	Wav file	**Duration**	**Transcribed(Copy pasted) text**	**Transcribed(Copy pasted) text**	**Duration**	**Wav file**
1	Sseg1.wav	00:04	ኢየሱስም በይሁዳ ቤተ ልሄም በንጉሱ በሄሮድስ ዘመን በተወለደ ጊዜ	ኢየሱስም በይሁዳ ቤተ ልሄም በንጉሱ በሄሮድስ ዘመን በተወለደ ጊዜ እነሆ ሰብአ ሰገል የተወለደው የአይሁድ ንጉስ ወዴት ነው	00:08	mat00.wav
2	Sseg2.wav	00:01	እነሆ ሰብአ ሰገል የተወለደው የአይሁድ ንጉስ ወዴት ነው	ኮከቡን በምስራቅ አይተን ልንሰግድለት መጥተናልና እያሉ ከምስ ራቅ ወደ ኢየሩሳሌም መጡ	00:06	mat01.wav
3	Sseg3.wav	00:01	ኮከቡን በምስራቅ አይተን ልንሰግድለት መጥተናልና እያሉ ከምስ ራቅ ወደ ኢየሩሳሌም መጡ ንጉሱ ሄሮድስም ሰምቶ ደነገጠ	ንጉሱ ሄሮድስም ሰምቶ ደነገጠ	00:03	mat02wav
4	Sseg4.wav	00:03	ኢየሩሳሌምም ሁሉ ከእርሱ ጋር የካህናትንም አለቆች የሕዝቡንም ጻፎች ሁሉ ሰብስቦ	ኢየሩሳሌምም ሁሉ ከእርሱ ጋር	00:03	mat03.wav
5	Sseg5.wav	00:01	ክርስቶስ ወዴት እንዲወለድ ጠየቃቸው እነርሱም አንቺ ቤተ ልሄም የይሁዳ ምድር	የካህናትንም አለቆች የህዝቡንም ጻፎች ሁሉ ሰብስቦ ክርስቶስ ወዴት እንዲወለድ ጠየቃቸው	00:07	mat04.wav

Table 3. 5 Manually and automatically (automatic I) segmented speeches in parallel

3.1.1.3. Graphemes Normalization

The other preprocessing task we made both on the training and test transcription was replacing characters whose sound same but have different shapes. This is because we want to check how the ASR performance changes while using the same characters on both the training and test transcriptions and vice versa. So in our case, we used graphemes (ሀሁሂሃሄህሆ) instead of using

(ሐሑሒሓሔሕሖሗ), and (ሰሱሲሳሴስሶሷ) instead of using (ሠሡሢሣሤሥሦሧ) because both have the same sound. To replace those graphemes we write a simple code/script with python.

Table3.6. depicts those used characters and replaced characters, used on both training and test transcriptions accordingly.

Used characters	**Replaced characters**
[ሀሁሂሃሄህሆ]	[ሃሐኀ],[ሑኁኹ],[ሒኂ],[ሓኃ],[ሔኄዄ],[ሕኅኽ],[ሖኆ]
[ሰሱሲሳሴስሶሷ]	[ሠሡሢሣሤሥሦሧ]
[ጸጹጺጻጼጽጾ]	[ፀ ፁ ፂ ፃ ፄ ፅ ፆ]
[ጕ]	[ኍ]
[አኡኢኣኤእኧ]	[ዐዑዒዓዔዕዖ]
Not considered or used	[ቐቑቒቓቔቕቖፗጟኻ ኹ ኺ ኻ ኼ ኸ ኸ ኾኳኵ]

Table 3. 6 Used and replaced characters both on the training and transcription files

3.1.2. ASR Development

According to the speech structure, three models were used in speech recognition to do the match. This includes an acoustic model, a phonetic dictionary, and a language model [**17**].

3.1.2.1. Phonetic Dictionary

The process of converting a target word from its written form (grapheme) to its pronunciation form (phoneme) is known as grapheme-to-phoneme (GTP) [**60**] . In the present study, GTP conversion was made based on phonetic dictionary used by Abate et. al. [**14**]. A total of 6,446 words with their respective phonemes were used for the ASR development using the bible data.

3.1.2.2. Language Model

The other component of ASR is the language model. We collected a total of 10,602 sentences from the Holy Bible [**61**] and WordProject [**59**] . We also used well-constructed sentences from [**14**] for our LVCSR development. Here, we tried to read available online resources and segmented in a way that will give meaning when it is read as a sentence. All unnecessary characters and punctuations are removed.

3.1.3. Speech Search and Locating Timeframe

In order to develop text-based STD, different researchers follow different architectures according to the development tools they used. To implement text-based STD, we followed general architecture from the literature [**33**] **.** The general architecture is already explained in the

literature review part of this report. One of the basic components of the text-based STD is the ASR. Because it requires conversion, the speech must be converted to its respective word. One of the main differences of text-based STD from other searching speech using text is that the system is not aware of the query term to be searched by users. This problem could be solved by developing LVCSR (ASR). In order to develop our ASR, we followed the same procedure we used for developing the ASR model for comparing automatically and manually converted speech (Bible ASR). The conceptual diagram in Figure 3.3., shows how the speech search (text-based STD) was developed.

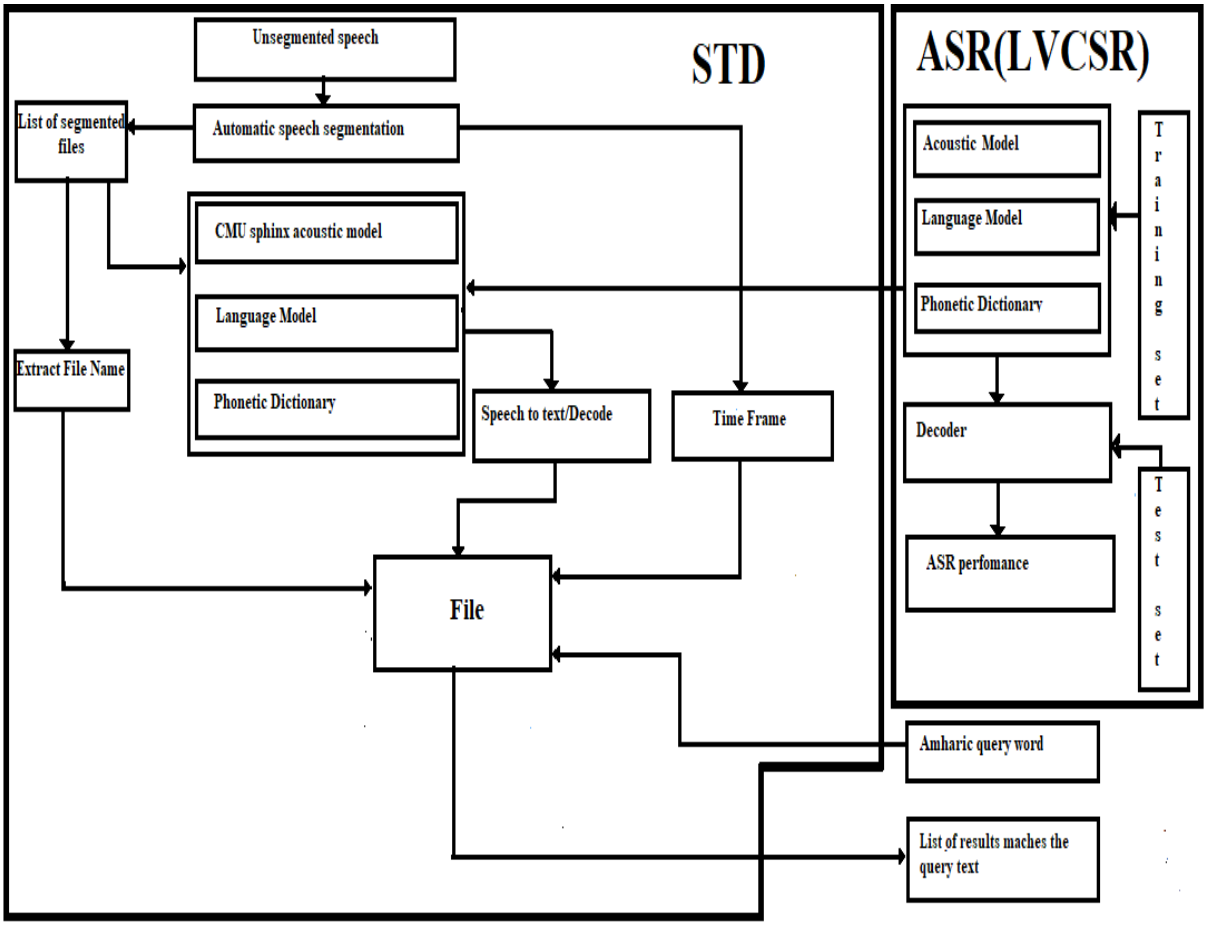

Figure 3. 3 Conceptual diagrams or workflow of how speech search system was developed

The diagram outlined in Fig 3.3., depicts how the speech search system was implemented. First, we developed two CMU Sphinx acoustic model using LVCSR from the broadcast and Bible domains. The first one was developed using LVCSR from the news domain. The other was developed by combining the LVCSR broadcast domain and the Bible domain. We used this

acoustic model and developed the txt-based STD system. On the text-based STD, speech search was begun from the unsegmented speech in a form of *.wav file. Then this unsegmented speech was segmented based on the threshold and minimum silence specified within the system. By using the acoustic model, Language model and phonetic dictionary and Sphinx4 API this automatically segmented speech was converted into text/decoding. The same segmented speeches file name was extracted and maintained. This made the alignment process easy. The other task was finding the time frame upon which the segmented speech was located. This was made by directly using the unsegmented speech and python's pydub package that continuously located time frame. Finally, file merging /alignment/indexing was made. Here we first combined the segmented file with the respective time frame and then using this result we combined it with the decoded text. All the generated files were maintained in windows file system. The aligned file maintained in windows file system contains, the segmented file name, in case users wanted to listen to only the segmented speech, location (start and end time), and decoded text. Therefore this file was easily searched and retrieved using a query word that would be given by the user. Details of implementation of each step are elaborated in the experimental design part of Chapter-4.

3.1.3.1. Corpus Preparation for LVCSR Training and Test Sets

Corpus is the basic component in LVCSR (ASR) development. The speech corpus we used for the development of an acoustic model was LVCSR sourced from published literature [**14**] Out of 100 speakers, we used 72 speakers read speech to train the acoustic model. The test speech was also directly used from the same source; i.e. LVCSR corpus. Therefore, by using the training and test speech corpus, we developed an acoustic model using CMU Sphinx toolkit. In addition, we combined the LVCSR training set with the Bible training set and developed ASR and checked its performance. The test sets/speeches we used for the merged training set were from the Bible domain (Bible test sets).

3.1.3.2. LVCSR Development

One of the problems of text-based speech search is that there is a large OOV rate. There is a lot of different approaches that are applied to tackle this problem. One of them is the use of very large decoding vocabulary.We have also used LVCSR developed by Abate et.al. [**62**] and a closed vocabulary language model. We developed two acoustic models. The first one was

developed using LVCSR broadcast domain only. The other was developed by combining the LVCSR broadcast domain and Bible speech.

3.1.3.3. Integration

The other component of text-based STD is the alignment. In this module of sub-component, different tasks are performed. The first task was the conversion of speech to its respective representative text and locating the time frame.

3.1.3.4. Testing and Evaluation of the STD System

The developed system was tested using unsegmented audio files from broadcast domain and Bible domain. For this, we recorded 8-minute speech using Samsung Galaxy J6+. The record was made in a silence home environment. The recorded text was taken from the test transcription of LVCSR corpus. This recorded speech was used by the system in a way that can be segmented, transcribed, and aligned with respected time frame and finally became ready for the search. Searching was made when the user give search query. The other speech we used to test the STD was from the bible domain where. The unsegmented speech was obtained from publicly available speech length 8 minutes and 16 seconds. Researches made on the STD are evaluated based on different set of words. One way of doing the evaluation is to randomly select limited number of most frequent words [**63**] [**64**] . We used Actual Term-Weighted Value (ATWV) to quantify the accuracy of text-based STD system we developed. ATWV we used for the evaluation could be found in chapter -5 of this report.

CHAPTER FOUR

EXPERIMENTAL DESIGN AND RESULTS

4. 1. Experimental Design

Before we execute our experiment, we set up tools for our experiment. The whole ASR experiment was conducted on Ubuntu 18.04 release and Intel® Core ™i7-6500U CPU @2.50 GHz 2.60 GHz RAM-8.00GB computer/laptop. For our ASR development, we used CMU Sphinx.

4.1.1 ASR Development

Generally, we developed two ASR systems. One was used for comparisons of automatically and manually segmented speech using the bible data. The other was developed using the LVCSR speech corpus to enable text-based STD development.

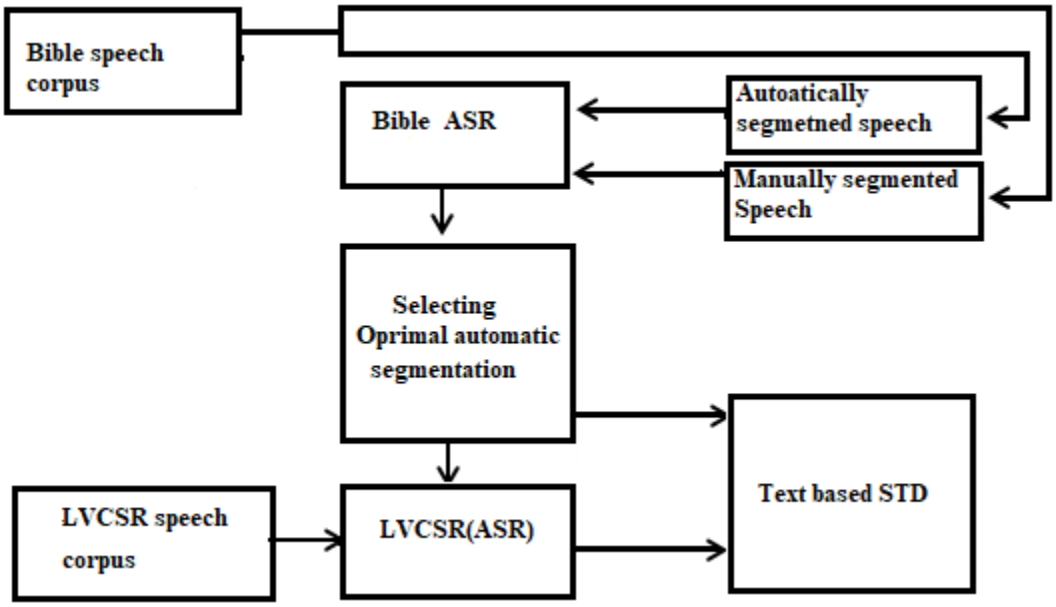

Figure 4. 1 General flow of the experiment for text-based STD development

Figure. 4.1. depicts the general flow that shows how the experiment was conducted. The first step was the development of ASR using bible speech corpus while the second step compared the performance and development of LVCSR that we used for the text-based STD.

4.1.1.1 Phonetic Dictionary

This was created by writing python script using the phonetic dictionaries sourced from published literature [**14**]. The phoneme for the word (ሀገር:country) are ሀ (h a) ገ (g aa) ር (r ee) and ሀሙስ (thursday) and ሀ(h a)ሙ(m u) ስ (s ee) using the above rules for every characters we convert word to its respective phone. Therefore, the phone for the word ሀገር and ሀሙስ will be (h a g aa r ee) and (h a m u s ee) respectively. Sample phonetic dictionaries are shown in appendix A of the report.

4.1.1.2 Language Model

To develop the acoustic model, the language model is important. We used SRI Language Modeling toolkit (SRILM) as a tool to create our language model. This is an open source tool that allows creating and testing various language models based on N-gram statistics [**65**]. Where N-gram is a form of probabilistic language model that uses the previous few words to predict the following item.The command we used to create a trigram model is shown below.

sudo ./ngram-count -order 3 -interpolate -text bibleText.txt -lm bible.lm.DMP

4.1.1.3. Training the Acoustic Model

We used the CMU Sphinx toolkit for our ASR development. To train our acoustic model we are required to prepare a file dictionary, phone, filler, language model, and transcription and file id. The findings are referred to as speech training databases. This database contains the information needed to extract the probabilities of a trained recording into an acoustic model [**66**].

- Pocketsphinx — recognizer library written in C.
- Sphinxtrain — acoustic model training tools
- Sphinxbase — support library required by Pocketsphinx and Sphinxtrain
- Sphinx4 — adjustable, modifiable recognizer written in Java

Therefore, we used the packages listed below to train our acoustic model:

- sphinxbase-5prealpha(required by sphinxbase and pocket sphinix)
- pocketsphinx-5prealpha(recognizer)
- sphinxtrain-5prealpha(acoustic model training tool)

By using the phonetic dictionary, training transcription, test transcription phones and the CMU **sphinxtrain run** command we generate the acoustic model [**17**]. The result of the command is shown in figure 4.2. Since we trained the acoustic model through varying training set, test sets and language model gave lot of results. However, after running the command, we have got the WER and sentence error rate as shown in the screenshot below Figure. 4.2.

```
MODULE: DECODE Decoding using models previously trained
        Decoding 87 segments starting at 0 (part 1 of 1)
        0%
        Aligning results to find error rate
        SENTENCE ERROR: 95.4% (83/87)   WORD ERROR RATE: 58.7% (458/780)
```

Figure 4. 2 Screenshot for the result of WER of the model after running the command

4.1.2 Text-based STD Development

All the system integration part of the STD system was done on Windows 10 pro. We used Netbeans 8.0.2 for java development and also implemented python on Anaconda environment.

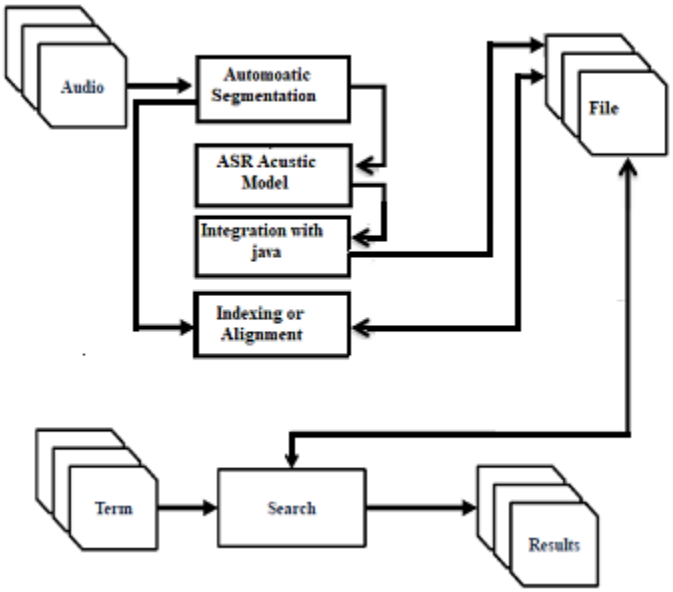

Figure 4. 3 System design architecture

The system design architecture shown in Figure. 4.3 depicts how the text-based STD was developed. The audio file was given to the system as input. Then, the speech was automatically segmented programmatically using python's pydub package. By using the acoustic model, which was developed using two acoustic models developed using LVCSR and ASR developed by

concatenating the LVCSR and Bible speech corpus, the segmented audio file was decoded and stored in a windows file system. The major task after decoding was the alignment process which includes time frame location. In this step, the time frame for every automatically segmented user's audio file was located and alignment of the transcription with the respective time frame was done. The transcription file was taken from the file and the list of segments with time frames are aligned and maintained in windows file system. Then the term was searched from the stored aligned file and the result was displayed. Details of the process are explained in the subsection of this section.

4.1.2.1. Unsegmented Speech (Audio)

The speech (Amharic audio speech) given to the text-based STD by the user is considered as an unsegmented speech. In the present research, we used unsegmented speech to evaluate our system. In order to check our text-based STD, we recorded an 8-minute read speech. The speech was recorded in a room environment. The speech was read by a woman in a way to simulate a news reading. The texts were then taken from the transcribed test texts from LVCSR [**14**] . The unsegmented speech was converted into *.wav since the ASR components of text-based STD could transcribe formats with *.wav.

4.1.2.2. Automatically Segmentation

This component of the STD system was used to segment the given unsegmented audio files into a list of segmented audio files. For a given unsegmented audio file based on the defined minimum silence and silence threshold, the system segments into slices /chunks. These slices/chunks were given iteratively to the ASR's acoustic model. Automatic segmentation was done by the function split_on_silence of python's pydub package.

4.1.2.3. ASR Acoustic Model

This model was used to decode or convert the speech into its respected text form using the automatically segmented speech developed in the above-mentioned step. So, the automatically segmented speech was an input for the acoustic model.

System integration uses different sub-component results. The first task to be done in system integration was to access the ASR's acoustic model inside java. This could help us to access all the transcription from GUI. For this, we used sphinx4 API which allowed us to access all the

acoustic models. Two ASR acoustic models were created and used. The first acoustic modesl was developed using only LVCSR. The other acoustic model was developed by combining the LVCSR and Bible speech corpus traning speech and LM of LVCSR and Bible. The LM was implemented by combining two language models (news, Bible) using the command shown in Figure 4.4. As shown in Figure 4.4 lambda values are values from [0.1-1] with the difference of 0.1 excluding 1 since the λ (lambda) value is between 0 and 1. The acoustic model we used for the STD development was LVCSR. We used files such as phonetic dictionary, language model.bin, where we convert it using the command (sphinx_lm_convert -i language_model.lm.(dmp) -o bin_language_model.lm.bin). Hence, the result was a language model in bin format. The other files we used for the development of the system (the acoustic model for transcription) were all inside filename.ci_cout like feat.params, mdef, mixture_weights, noisedict, transition_matrices, and variances. By using these files and sphinx4 API incoming segmented *.wav files could be recognized or transcribed. The system integration of ASR with java was made by following the procedures described on the sphinx site [**17**].

```
getnet@ubuntu:~/Desktop/srilm/bin/i686-m64$ sudo  ./ngram -order 3 -lm intpolNew
s.lm -mix-lm intpolBible.lm -lambda 0.1 -write-lm intepBibleNews.lm
```

Figure 4. 4Interpolating two language models using news and Bible LM

4.1.2.4. Locating the Time Frame

To locate the time frame, different steps were followed. The time frame was located using python's pydub library . The function we used called "detect_nonsilent", and this function locates the time frame by first detecting non silent part of the speech or using the spoken parts of the speech. This segmentation was performed using parameter like "min_silence_len" and "silence_thresh". So, we wrote a code using python with the script named "timeframe.py" which could locate the time frame of the segmented speech. The last step of the process was alignment. In this step, the time frame and transcription made by ASR were aligned which is shown in Appendix E of the report.

4.1.2.5. Integration (Alignment)

Once the segmented speech is decoded and maintained in a text file, the next step is an alignment. In this step, we find and map the time upon which transcribed or decoded text is available. All these were done using java and python in combination. Using java, we called

programmatically every python scripts and sphinx 4 acoustic models. Here, the time frame for every segmented speech was done by python's pydub package. The final step was just searching a word from a transcribed text file using text query. We tried to append the time frame with respective segmented audio files.

We used Netbens IDE to integrate all the sub-components. We further used sphinx 4 API inside java and the python scripts. In addition, we developed GUI using javas JFrame which is a top-level container that provides a window on the screen. Sample code inside Netbeans is shown in the Appendix H.

4.1.2.6. Searching

The final step of the STD was query term searching. This was done using the aligned text file stored in the Windows file system. We used simple java string mapping function which compares query terms and the aligned text and retrieve the time with the transcribed text. The Sample GUI code we used for the GUI development is shown in the Appendix H of the report.

4.2. Experimental Results and Discussion

In this section, the results obtained from our experiment are discussed. This includes comparison of the manually and automatically segmented speech as well as text-based STD that we developed.

4.2.1. Test Speech Comparison

Discussion on comparison of automatically and manually segmented speeches is presented in this section. Changing the training, test and language models have impact on the ASR system that we developed using Bible speech corpus.

4.2.1.1. Comparison between Automatically and Manually Segmented Speech Varying LM

The evaluation is made using the popular ASR evaluation metrics WER. Apart from performance comparison for training transcription, this is used to compare the evaluation for the ASR test speech.

Column Name Description

The description of each column we used that shows the experimental result on Table 4.1 described here. The column named, **Model id** was given by the authors for ASR model we generated where the model id **RM/RA/RSA OV-xx** (ResultManual, ResultAutomatic, ResultAutomoaticSmall-Open Vocabulary and the final label is number of sentences we used while generating the language model) and **RM/RA/RSA CV-xx** (ResultManual, ResultAutomatic, ResultAutomoaticSmall-Close Vocabulary and the final label is number of sentences we used while generating the language model), **No of training sentences for LM** (number of sentences used to create a language model), Training set (data used to train the recognizer), **Manual(test set)** (manually segmented test speech), **Automatic I** (automatically segmented test set/speech more like phrase and word like segmentation), the **Automatic II** (sentence like segmentation or whose segmentation is long) and column (**Manual- Automatic I**) is the difference between the manually segmented speech with that of the automatically segmented speech I). The last column (**Manual-AutomaticII**) shows the difference between the manually segmented speech and the automatically segmented speech II or the result of phrase-like segmentation. For example, RM/RA/RAS OV-5000 refers id for RM OV-5000, RA OV-5000, and RAS OV-5000 manually, automatically and automatically with phrases or word like segmentation respectively.

No				Test transcriptions sentences			WER (%)			Difference	
	Model id	No of training sentences for LM	Training set	Manual(test set)	Automatic I	Automatic II	Manual	Automatic I	Automatic II	Manual-Automatic I	Manual-Automatic II
Open Vocabulary											
1	RM/RA/RAS OV-5000	5000	1-950	90	91	245	78.3	81.1	91.1	2.8	10
2	RM/RA/RAS OV-10000	10000	1-950	90	91	245	76.8	79.0	89.6	2.2	10.6
3	RM/RA/RAS OV 10600	10600	1-950	90	91	245	71.0	74.1	87.1	3.1	13
Close Vocabulary											
1	RM/RA/RAS CV-5000	5000	1-950	90	91	245	66.8	68.1	82.0	1.3	15.2
2	RM/RA/RAS CV-10000	10000	1-950	90	91	245	63.9	66.0	81.4	2.1	17.5
3	RM/RA/RAS CV 10600	10600	1-950	90	91	245	54.4	56.5	77.6	2.1	23.2

Table 4. 1 The effect of language model on the ASR comparison performance

The tabulated results (Table 4.1) show mainly the effect of changing language model on the manually and automatically segmented speeches. For the comparison, we used the same length of speech.

The first three experiments showed how the training text used in developing the language model positively affects the WER.

We found that increasing the size of training text for the development of the language model to 10.6k sentences increases the performance of the ASR in all the systems. Here, the manually segmented speech and automatically segmented speech (Automatic I) show closer results. It also

shows a higher WER difference when compared to Automatic II where the segmentation is most like phrase and word level. In addition, using closed vocabulary in the language model showed a major decrease in the WER and increase in the ASR performance when it is compared with using open vocabulary.

The three screenshots in Figure 4.5, 4.6 and 4.7 shows the perplexity for language models obtained using the developed using 5, 000, 10,000 and 10,602 sentences respectively. The result was obtained using the command

sudo ./ngram-count -order 3 -interpolate -text lm.txt -lm bible.lm -vocab bible.vocab

sudo ./ngram -lm bible.lm -ppl bibleTest.test

```
file bibleTest.test: 90 sentences, 803 words, 0 OOVs
0 zeroprobs, logprob= -3047.18 ppl= 2583.99 ppl1= 6233.61
```

Figure 4. 5 Perplexity of the language model using training text of 5,000 sentences

```
file bibleTest.test: 90 sentences, 803 words, 0 OOVs
0 zeroprobs, logprob= -2910.93 ppl= 1818.53 ppl1= 4217.63
```

Figure 4. 6 Perplexity of the language model using training text of 10,000 sentences

```
file bibleTest.test: 90 sentences, 803 words, 0 OOVs
0 zeroprobs, logprob= -2547.23 ppl= 711.943 ppl1= 1486.44
```

Figure 4. 7 Perplexity of the language model using training text of 10,602 sentences

In addition to the perplexity of the LM's we also computed the OOV rate and vocabulary size for the training text of 5, 000, 10,000 and 10,600 sentences.

No				No. language test sentences			OOV rate		
	LM	No of training sentences for LM	No of unique vocabularies	Manual(test set)	Automatic I	Automatic II	Manual	Automatic I	Automatic II
1	OOV5000LM	5000	13,573	90	91	245	218	219	222
2	OOV10000LM	10000	25,673	90	91	245	153	154	158
3	OOV10600LM	10600	26,364	90	91	245	100	101	103

Table 4. 2 OOV rate and unique vocabularies of the language model using manual, automatic I and automatic II as a test text

Table 4.2 shows the OOV rates of language model that was trained with different training (5000,1000,10600) texts .The result shows as the size of the vocabulary increases the OOV rate decreases and the better the language model. Therefore, the size of the vocabulary increases the perfomace of ASR.

4.2.1.2. Performance Comparison Using Training Speech dataset

Performance of ASR system was compared by varying the training sets holding test transcriptions same in every iteration. This was conducted to check how the ASR performs while changing the training set and its effect on the automatically segmented speeches.

No			Test transcriptions sentences			WER (%)			Difference	
	Model Id	Training set	Manual(test set)	Automatic I	Automatic II	Manual	Automatic I	Automatic II	Manual-Automatic I	Manual-Automatic II
1	RM/RA/RSA-100	100	90	91	256	64.8	65.1	84.8	0.3	20
2	RM/RA/RSA-200	200	90	91	256	59.9	58.7	81.6	1.2	21.7
3	RM/RA/RSA-300	300	90	91	256	56.4	58.8	82.0	2.4	26.5
4	RM/RA/RSA-400	400	90	91	256	54.9	59.1	81.5	4.2	26.6
5	RM/RA/RSA-500	500	90	91	256	54.5	59.8	81.3	4.9	26.8
6	RM/RA/RSA-600	600	90	91	256	55.9	60	81.2	4.1	25.3
7	RM/RA/RSA-700	700	90	91	256	55.3	56.8	81.5	1.5	26.2
8	RM/RA/RSA-800	800	90	91	256	56.9	57.5	81.6	0.6	24.7
9	RM/RA/RSA-900	900	90	91	256	54.4	58.2	83	3.8	28.6
10	RM/RA/RSA-1000	1000	90	91	256	55.5	56.5	82.2	1	26.6
11	RM/RA/RSA-1050	1050	90	91	256	55.2	58.9	82.5	3.7	27.2

Table 4. 3 Result of manually segmented speech through using different sets of training data

Column Name Description

The description of the each column we used that shows the experimental result on Table 4.3 described here.The column named, **Model id** used to show the result obtained by varying the training sets. For example RM/RA/RSA-100 refers id for RM-100, RM-100, RM-100 manually, automatically and automatically with phrases or word like segmentation respectively using 100 sentences in training set. Therefore RM/RA/RSA-200 to show we used 200 training senteces. Column **Training set** (data used to train the recognizer), **Manual (test set)** (manually

segmented test speech), **Automatic I** (automatically segmented test set/speech more likely phrase and word like segmentation), the **Automatic II** (sentence like segmentation or whose segmentation is long) and column (**Manual- Automatic I**) is the difference between the manually segmented speech with that of the automatically segmented speech I). The last column shows the difference between the manually segmented speech and the automatically segmented speech II or the result of phrase-like segmentation.

Table 4.3 shows how the performance of ASR varies while changing the training set, holding the language model and test speech the same and using close vocabulary. The total sentences used to create a language model were 10602. All characters used in the test transcription were also being available on the training set.

From the above result, Automatic I column and Manual column shows very slight difference in WER. As the WER of the manual decreases, the WER result of the automatic I also decrease, and vice versa. However, the WER result of Automatic II was very large when compared to that of the manually segmented test speech in every experimental iteration. For example in the 1st iteration using 100 training sentences we found a WER of 64.8, 65.1 and 84.8 respectively having a WER difference of 0.3 and 20 for Manual-Automatic I and Manual-Automatic II respectively.

This implies sentence level automatic segmentation (Automatic I) would yield very close result to manually segmented speech. On the other hand, Automatic II shows very large difference with the manually segmented speech in every experimental iteration. This is because the language model performs better if the segmentation is sentence-like unit apart from word or phrase level segmentations [**67**].

4.2.1.3. Performance Comparison Using Test Sets

In this experiment, comparison was made to check the performance of ASR by varying the length of test transcription holding the training corpus same in every experimental iteration. In the experiment even if length of sentence is varied, content of the speech (starting and ending) of the speech is same for test transcription.

No			Test transcriptions sentences			WER (%)			Difference	
	Expermental result id	Training set	Manual(test set)	Automatic I	Automatic II	Manual	Automatic I	Automatic II	Manual-Automatic I	Manual-Automatic II
Before graphemes normalization										
1	RM/RA/RSA-C1	1050	60	60	167	69.8	75.9	87	6.1	17.2
2	RM/RA/RSA-C2		87	91	245	74.9	75.9	88.2	1.0	13.3
3	RM/RA/RSA-C3		105	111	295	76.3	78.2	89.3	1.9	13
4	RM/RA/RSA-C4		214	248	602	74.4	78.5	91.1	4.4	16.7
5	RM/RA/RSA-C5		307	369	877	74.5	79.3	94.1	4.8	19.6
After graphemes normalization										
6	RM/RA/RSA-C6	1050	60	60	167	69.8	74.6	86.1	4.8	16.3
7	RM/RA/RSA-C7		87	91	245	73.0	74.1	87.1	1.1	14.1
8	RM/RA/RSA-C8		105	111	295	74.6	76.8	88.3	2.2	13.7
9	RM/RA/RSA-C9		214	248	602	73.2	77.6	90.6	4.4	17.4
10	RM/RA/RSA-C10		307	369	877	73.5	78.5	93.6	5.0	20.1

Table 4. 4 The impact of ASR by maximizing length of test speech holding the contents of manual, automaticI, automatic II same

Column Name Description

The description of each column we used that shows the experimental result on Table 4.4 described here.The column named, **Experimental result id** used to show the result obtained by varying the test sets. For example RM/RA/RSA-C1 Result of Manual/Automatic(sentence like segmentation)/SmallAutomatic(phrase like segmentation) and C1 which was segmented in 60,60 and 167 respectively. In the same way the second segmentation RM/RA/RSA-C2 uses 87(Manual segmentation), 91(sentence like segmentation), 245(word or phrase like segmentations) where the length and content of the audio file was the same except varied segmentations 87, 91 and 245.

The result shown in Table 4.4.was based on experiment done using the open vocabulary language model and before graphemes normalization and after graphemes normalization of the training and test sets. In addition, since we didn't incorporate all words which sounds the same, different representation like replacing ሐ,ኻ (h a) with ሀ (h a) shows a decrease in word error rate from 74.9 to 73.0 in the 2nd and 7th iteration, 76.3 to 74.6 in the 3rd and 8th iteration and all the remaining iterations such as 4th and 9th,5th and 10th also shows a decrease in WER except on the 1st and 6th iteration which shows the same result using the manually segmented speech. For automatic I, we also found a decrease in WER difference from 75.9 to 74.6, 75.9 to 74.1, 78.2 to 76.8, 78.5 to 77.6, 79.3 to 78.5 in the 1st and 6th, 2nd and 7th, 3rd and 8th, 4th and 9th, 5th and 10th iterations respectively. For the automatic II, similarly, we found a decrease in WER difference from 87 to 86.1, 88.2 to 87, 89.3 to 88.3, 91.1 to 90.6, 94.1 to 93.6 in the 1st and 6th, 2nd and 7th, 3rd and 8th,4th and 9th, 5th and 10th iterations.

In general in this experiment, we found that using same graphemes both on the training and test sets shows a slight decrease in the word error rate of both on the manually and automatically segmented speeches.

In general, in this experiment, we experimented the impact of increasing both training and test speech corpus. In addition, we used graphemes, closed vocabularies, which decrease the WER and compared the result of the manually and automatically segmented speech. In all experiments, the sentence-like segmentation (Automatic I) shows a closer WER difference to that of the manually segmented speech when compared to Automatic II which shows a relatively high WER difference.

Therefore from this experiment we answered the research question "*What the effect is of manually and automatically segmented test speech on ASR performance*".From the result we found manually segmented test speech showed less WER. The automatically segmented test speech (sentence like segmentation/Automatic I) showed a less WER or closer to the manually segmented test speech. On the other hand (phrase/word like segmentation) showed a larger WER compared to both the manually and automatic I (sentence like segmentations).

4.2.1.4. ASR Performance Comparison Using Different Domains

This section shows the performance of the recognizer (LVCSR) upon using speech corpus developed using the news speech as training and Bible and news as tested speech corpus.

No	Training speech	Test speech	LM	WER (%)
1	News speech corpus	Bible speech corpus	Open vocabulary	92.8
2		News speech corpus	Open vocabulary	64.9
3		Bible speech corpus	Close vocabulary	90.6
4		News speech corpus	Close vocabulary	53.6

Table 4. 5 Performance of ASR upon using different test speech domain corpus

The data shown in Table 4.5 shows the performance of LVCSR [**14**] upon using different test speech domains. We used an 8-minute test speech with different speech content. Since the LVCSR trained with news speech corpus the error rate of the recognizer is less compared with that of the bible test speech corpus which is a different domain. Even if using a closed vocabulary decreases the WER, we didn't observe a significant difference between the two domains. Therefore from the result, we selected a minimum WER for our system development. We developed a system with LVCSR's WER of 53.6 which has the best performance.

4.2.1.5. ASR Performance Combining LM and Different Domain speech

This section shows the performance of the ASR obtained by combining training speech corpus and LM of two different domains. The first domain was from the news LVCSR speech corpus and the other was the Bible speech corpus. This experiment was conducted because the WER of ASR was 90.6% as depicted in Table 4.5, upon using Bible test speech corpus on ASR trained using LVCSR news speech corpus. Therefore we conducted another experiment to check the performance of ASR by combining the LVCSR and Bible training speech corpus and LM of the LVCSR and the Bible.

Training speech	Test speech	λ (Lambda) Bible	WER%
Bible+News speech corpus	Bible speech	0.1	46
		0.2	46
		0.3	46
		0.4	46
		0.5	46
		0.6	46
		0.7	46
		0.8	46
		0.9	46
		1	47

Table 4. 6 Using interpolation technique to improve the WER of the speech recognizer

The data shown in Table 4.6 .shows WER of a speech recognizer developed using LVCSR developed by combining LVCSR news speech corpus and bible speech corpus. The result was obtained by changing lambda values with an interval of 0.1. We obtained a result of 46 % for all experiments except for one we found 47 % where the lambda value was 1.

From the two experiments we made, we found that combining the LVCSR with the Bible corpus and interpolating the LM developed for LVCSR and Bible speech corpus shows a decrease in WER of the ASR. From Table 4.4, WER of 90.6% was obtained by using LVCSR news speech and bible as a test speech. By using the same Bible test speech but by combining the two different domains of news and Bible speech and interpolating the LM's of LVCSR news and Bible LM's we found a WER of 46 shown in Table 4.5, which shows a large WER difference of 44.6. Therefore combining two different training speech corpus and LM's, would yield a decrease in WER or a better ASR model. We also used this result (acoustic model of ASR with WER of 46 %) and developed a text-based STD which is from the Bible domain.Therefore the research question "*What is the effect of ASR recognition errors on searching using different domains*", was answered as showen in the tabulated result of Table 4. 7 where using same domain to train and to test the performance of ASR shows a better WER that using different domaing i.e bible domain for ASR which was developed using broadcast speech domain.In addition we combined the traning speech corpus of the LVCSR and Bible speech corpus and interpolate the langage models of the LVCSR and the bibe speech and we experimented it using the bible speech corpus which we found a slight decrease in the WER.

4.2.1.6. Segmentation Time

While performing preprocessing or transcription, we tried to manually segment or trim for some portion of the unsegmented file using the most frequently used software audacity. It took an average time of 3 hours to segment a 3-minute and 30-second speech into 40 chunks or segments. The task in audacity included naming the file according to user's interest which may result in error and tedious task. This time doesn't include the transcription part which is listening and copying and pasting text using different editors because it is a common task that is done in either of the two (manual and automatic transcription).

In order to have a full audio file, we manually segmented for at least 3-minute and 30-second using Audacity. Apart from performance comparison, we also tried to check for the time taken to segment unsegmnted audio file even if it is short duration in length. While performing automatic segmentation, we used Audacity software which is audio processing software. For a 30minute and 30-second file, it took an estimated 30-45 minutes by the researcher. The result can be used as a baseline to forecast when the audio file is large. However, the time it took us to segment the unsegmnted speech using pythons pydub package for an audio file length (1:15:35) was 0.0:8.0:2.9559926986694336(zero hour, 8 minute and 3 seconds) which is less compared with the manually segmented speech.

4.2.2. Text-based STD Development

In this experiment, we try to design and developed a text-based STD and evaluated the performance of the system using ATWV (Actual Term Weighted Value).

4.2.2.1. Speech Search using News Data

The screenshot shown in Figure 4.8 depicts Amharic text-based STD GUI (Graphical User Interface) which can accept Amharic query text (from news domain) and display where the speech is located in hour, minute and seconds. Starting and ending times are displayed within the square bracket as shown in the text area of text-based STD GUI.

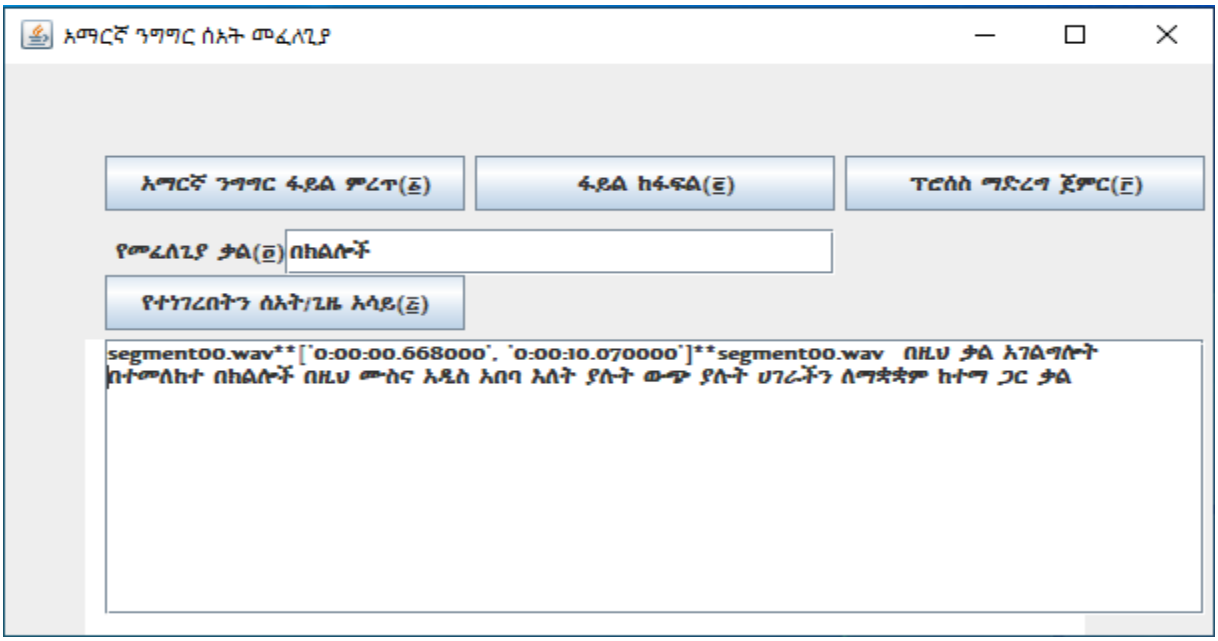

Figure 4. 8 GUI for searching the speech using news query word

The following steps are to be followed to search a speech and then the system to locate a time frame for the spoken speech in the file. The user first selects an audio file over which speech is to be searched. To select a file from a computer system, user can select a button labeled አማርኛ ንግግር ፋይል ምረጥ(፩) (select speech file (1)) where geez ፩(One) or 1 the GUI that shows list of directories after selecting the button is shown in appendix J of the report, it is a first step that a user has to use. After the user selects a file, the other button labeled (ፋይል ከፋፍል(፪)) or segment file(2) where the geez ፪ (two) or 2, need to be selected. In this step, the system will internally segment the audio file which is already selected in the first step. Segmentation of the audio file is made automatically according to the parameter given or using a minimum silence and silence threshold. After segmentation is completed, the third step is the basic process where the time frame of every segmented speech is located or identified, and transcription and alignment tasks are made. The fourth step is requiring users to insert an Amharic query text የመፈለጊያ ቃል(፬) or search query(4). Then the final step is getting the result from a system; i.e., getting the time frame of spoken speech through simple searching that is made using java code. Finally, when the user presses the button with the label የተነገረበትን ሰአት/ጊዜ አሳይ(፭) show the time(5), the system will show the time interval over which the query text is located through tracking and searching from the selected audio file which was given in step1. Sample result is shown in the text area of Figure 4.8.

Query term	Transcription should be	Transcription by the system	Correctly transcribed and located
በክልሎች	segment00.wav [የኢንተርኔት አገልግሎትንም **በተመለከተ በክልሎች** በዞኖችና በአዲስ አበባ በተለያዩ ቦታዎች የአገልግሎት ማእከሎችን **ለማቋቋም** መታቀዱን አብራርተዋል]	segment00.wav** ['0:00:00.668000', '0:00:10.070000']** segment00.wav በዚህ ቃል አገልግሎት **በተመለከተ በክልሎች** በዚህ ሙስና አዲስ አበባ እለት ያሉት ውጭ ያሉት ሀገራችን **ለማቋቋም** ከተማ ጋር ቃል	**በተመለከተ** **በክልሎች** **ለማቋቋም** የአገልግሎት/ **አገልግሎት**

Table 4. 8 Search result description using news data

The above table (Table 4.7) depicts that the systems transcription, transcription by hand and words which are correctly recognized by the system. Manual and system transcriptions are highlighted in bold to show the difference.

The column 'Transcription should be' (Table 4.7), it shows what it should look like if it is transcribed without error. The result "segment00.wav [የኢንተርኔት አገልግሎትንም **በተመለከተ በክልሎች** በዞኖችና በአዲስ አበባ በተለያዩ ቦታዎች የአገልግሎት ማእከሎችን **ለማቋቋም** መታቀዱን አብራርተዋል]" is a manually transcribed speech and segment00.wav is its respective audio file. The column "Transcription by the system*"* (Table 4.7), shows how the system decodes and locates its respective time frame. From the result "segment00.wav** ['0:00:00.668000', '0:00:10.070000']**segment00.wav በዚህ ቃል አገልግሎት **በተመለከተ በክልሎች** በዚህ ሙስና አዲስ አበባ እለት ያሉት ውጭ ያሉት ሀገራችን **ለማቋቋም** ከተማ ጋር ቃል *",* segment00.wav is the segmented audio file where the query text (በክልሎች) is located. The time in square bracket ['0:00:00.668000', '0:00:10.070000'] shows the interval upon which the query text በክልሎች is located. The square bracket '0:00:00.668000' indicates the search query term begins at 0 hours 00 minutes and 00 seconds whereas '0:00:10.070000' indicates the search query term ended at 0 hours 0 minute and 10 seconds. We can say that the word **በክልሎች** is located in between 0 seconds and 10 seconds of the selected audio file. Sample system transcription and respective time frames for the 7-minute and 59 seconds test speech could be found in appendix F of the report. The last column of Table 4.7 shows a list of words **በተመለከተ**, **በክልሎች**, **ለማቋቋም** which are correctly recognized by the system and if these terms are to be searched, their time frame can be correctly located by the system. The other word **የአገልግሎት** was

retrieved as **አገልግሎት** where the prefix **የ** was missing, which could be solved using Amharic steamers which is not included in this study. However, if we search for the word **የኢንተርኔት** as shown in Figure 4. 9, we could not find it at segment 00 (between 0 and 10 seconds) because it is not correctly recognized by the system. This requires further improvement of the ASR. Therefore, if we can develop and have a better speech recognition system, decoding and system retrieval for every query term in a given speech could be located with its respective time frame more precisely. Therefore the research question, "*What is the effect of ASR recognition errors on searching the recognized text?"* was answered since ASR didn't recognize/decode all words correctly we were not able to get the word **የኢንተርኔት** even if this word was spoken in the audio file.

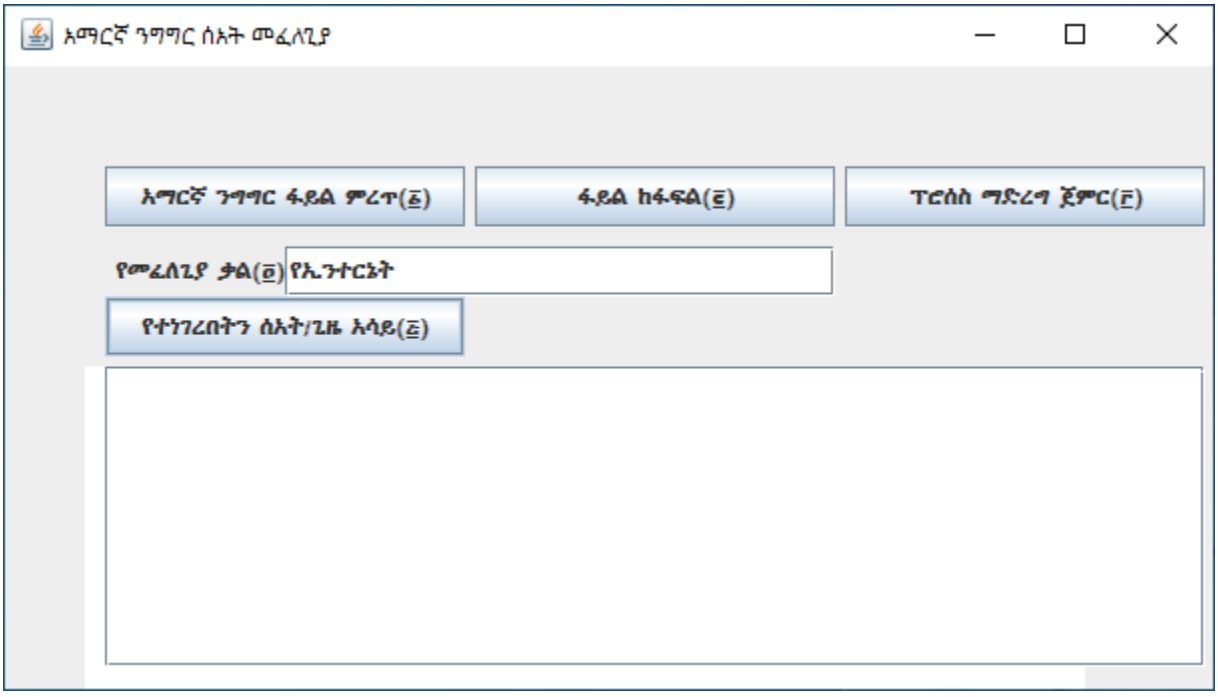

Figure 4. 9 GUI for searching the speech using different news query word

4.2.2.2. Speech Search using Bible Data

This section shows text-based STD which was developed using ASR with a WER of 46% tested with a bible speech length of 8 minutes and 16 seconds. The result was obtained by interpolating the language models of LVCSR and Bible language models. The screenshot shown in Figure 4.10. depicts Amharic text-based STD GUI (Graphical User Interface) which can accept

Amharic query text (from Bible domain) and display where the speech is located in hour, minute and seconds.

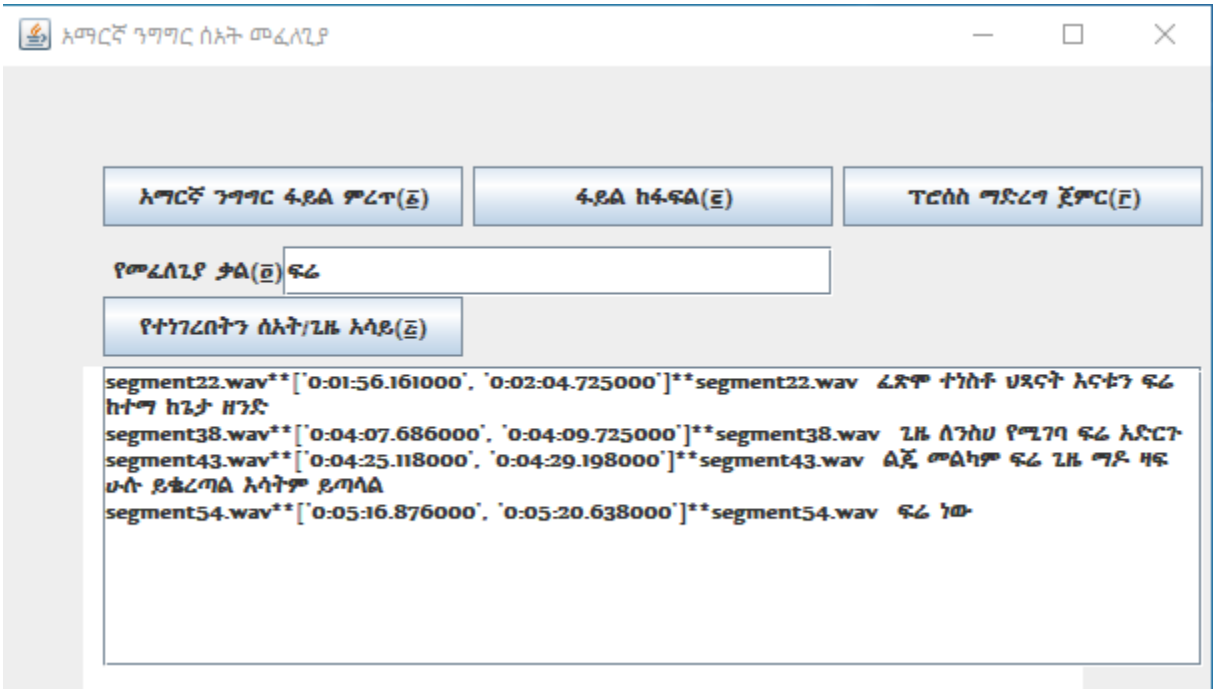

Figure 4. 10 GUI for searching the speech using Bible-related query word

A detailed explanation of the retrieved result shown in Figure 4.10, obtained by using the query term ፍሬ is shown in Table 4.8. From Table 4.8 segment 22 and segment 54, the search term ፍሬ was incorrectly transcribed by ASR which was not spoken at segment 22 and segment 54 on the time interval [0:01:56-0:02:04] and [0:05:16-0:05:20] respectively. In addition, no word was correctly recognized in segment 54 compared to all other segments. However, on segment 38 and segment 43, the word ፍሬ was correctly transcribed and the query term could be found at the time specified by the system. In addition at segment 22, the word ሀጻኑንና was decoded as ሀጻናት which could be solved by finding the root word (steamer). Moreover, the word ፍሬ has appeared two times and the system located and displays two search results in every segment (segment 38 and 43) it was correctly decoded by the ASR. Sample system transcription and respective time frames for the 8-minute and 31 seconds Bible test speech could be found in appendix G of the report.

Query term	Transcription should be	Transcription by the system	Correctly transcribed and located
ፍሬ	segment22.wav [እርሱም **ተነስቶ** ህጻኑንና **እናቱን** በሌሊት ያዘና **ከጌታ ዘንድ** በነቢይ ልጇን ከግብጽ ጠራሁት የተባለው እንዲፈጸም ወደ ግብጽ ሄደ]	segment22.wav**['0:01:56.161000', '0:02:04.725000']**segment22.wav ፈጽሞ **ተነስቶ** ህጻናት **እናቱን** ፍሬ ከተማ **ከጌታ ዘንድ**	**ተነስቶ** **እናቱን** **ከጌታ** **ዘንድ** ህጻኑንና/ህጻናት
	segment38.wav [እንግዲህ **ለንስሀ የሚገባ ፍሬ አድርጉ**]	segment38.wav**['0:04:07.686000', '0:04:09.725000']**segment38.wav ጊዜ **ለንስሀ የሚገባ ፍሬ አድርጉ**	**ለንስሀ** **የሚገባ** **ፍሬ** **አድርጉ**
	segment43.wav [እንግዲህ **መልካም ፍሬ** የማያደርግ **ዛፍ ሁሉ ይቁረጣል** ወደ እሳትም **ይጣላል**]	segment43.wav**['0:04:25.118000', '0:04:29.198000']**segment43.wav ልጇ **መልካም ፍሬ** ጊዜ ማዶ **ዛፍ ሁሉ ይቁረጣል እሳትም ይጣላል**	**መልካም** **ፍሬ** **ዛፍ** **ሁሉ** **ይቁረጣል** **እሳትም** **ይጣላል**
	segment54.wav [እነሆም ሰማያት ተከፈቱ የእግዚአብሄርም መንፈስ እንደ ርግብ]	segment54.wav**['0:05:16.876000', '0:05:20.638000']**segment54.wav ፍሬ ነው	No word was recognized

Table 4. 9 Search result description using Bible data

4.2.3. Performance Evaluation of STD

The performance of text-based STD is evaluated using the most frequent and less frequent words on the speech corpus. Its accuracy could be measured using Actual Term-Weighted Value (ATWV) [**68**]. The ATWV evaluation is performed on an 8-minute read speech which was recorded by us as explained in the methodology part of this report.

4.2.3.1. Performance Evaluation (ATWV)

Actual Term-Weighted Value (ATWV) is a new metric, created to reflect one potential use of an STD system. It is used to quantify system accuracy on a particular set of query words [**68**].

ATWV is defined in equation 5.1 as:

$$\text{ATWV} = \text{mean}\left(\frac{N_{\text{correct}}(s)}{N_{\text{true}}(s)} - \beta \cdot \frac{N_{\text{spurious}}(s)}{T - N_{\text{true}}(s)}\right) \quad (5.1)$$

Where the search term s occurs Ntrue(s) times in the reference transcript and the system makes Ncorrect(s) correct and Nspurious(s) incorrect assertions of s. T is the total duration of the audio corpus in seconds. The parameter β incorporates the relative costs of misses and false assertions and the prior probabilities of search terms; it was set to 999.9 for the evaluation. To avoid division by zero, the mean is taken over only the terms in the set for which Ntrue(s) is positive.

4.2.3.2 Selected Words for Evaluation using News Speech

To measure the accuracy, we selected a total of 26 most and less frequent words within a training set of LVCSR. Most frequent words are words whose frequency is greater than 60 or words that exist more than 60 times and less frequent words are those which exist less than 30 times in the training set of the speech corpus. A list of all words with their frequencies is shown in appendix D of the report.

No	Type	Word	Frequency in training corpus	Ntrue	Ncorrect	Nspurious	ATWV
1	Most frequent word	አዲስ	235	2	1	1	0.46
2		ዛሬ	194	1	1	0	1
3		ቤት	190	2	1	1	0.46
4		አቶ	167	3	1	2	0.25
6		ክስ	143	2	1	1	0.46
7		አበባ	113	1	1	0	1
8		ከተማ	112	1	1	0	1
9		ስብሰባ	112	1	1	0	1
10		ተጨማሪ	101	1	1	0	1
11		አራት	99	1	1	0	1
12		ሜዳ	74	1	1	0	1

13		ስምንት	60	1	1	0	1
14	Less frequent word	ሽልማት	25	1	1	0	1
15		ሰኞ	22	1	1	0	1
16		ዘጠና	22	1	1	0	1
17		ማህበር	21	1	1	0	1
18		ቀይ	16	1	1	0	1
19		ህግና	15	1	1	0	1
20		ዘመን	12	1	1	0	1
21		መኖሪያ	8	1	1	0	1
22		የስፖርት	7	1	1	0	1
23		የኳስ	2	1	1	0	1
24		በጸረ	4	1	1	0	1
25		በቀጥታ	4	1	1	0	1
26		በክልሎች	1	1	1	0	1
Average							0.98

Table 4. 10 Most and less frequent news words

AWTV➔ (98%)

Duration of test speech➔ (7:59)

WER ➔(53.6)

The results in Table 4.9 show the performance of text-based STD upon using ASR which was developed using LVCSR with ATWV of 98.6%. In order to test the system, we used a 7-minute and 59-second test speech as described in the methodology section.

4.2.3.3. Selected Words for Evaluation using Bible Speech

In order to measure the accuracy, we selected a total of 26 most and less frequent words within a training set of LVCSR. Most frequent words are words whose frequency are greater than or equal to 15 or words exist more than 15 times and less frequent words are those which exists less than 15 times in the training set of the speech corpus. List of all bible words with their frequencies are shown in appendix E of the report.

No	Type	Word	Frequency in training corpus	Ntrue	Ncorrect	Nspurious	ATWV
1	Most frequent word Above >=15	ሆነ	84	2	1	1	0.46
2		አራት	83	1	1	0	1
3		መንግስታት	44	1	1	0	1
4		ልጆች	31	2	1	1	0.46
6		ወራት	25	1	1	0	1
7		ቅዱስ	22	1	1	0	1
8		እጅግ	21	3	2	1	0.57
9		ደስ	21	4	3	1	0.7
10		ህጻናት	20	1	1	0	1
11		ምእራፍ	19	2	1	1	0.46
12		ዘመን	19	4	2	2	0.49
13		መልካም	19	1	1	0	1
14	Less frequent word=<15	ፍሬ	10	2	2	0	1
15		ክርስቶስ	8	1	1	0	1
16		መላእክት	8	1	1	0	1
17		ዛፍ	4	1	1	0	1
18		መጥምቁ	3	1	1	0	1
19		ተራራ	3	1	1	0	1
20		ከማርያም	1	1	1	0	1
21		በመቅደስ	1	1	1	0	1
22		ለንስሀ	1	2	2	0	1
23		ኮከብ	1	1	1	0	1
24		በራማ	1	1	1	0	1
25		ጠፍር	1	1	1	0	1
26		ጠጉር	1	1	1	0	1
Average							0.89

Table 4. 11 Most and less frequent Bible words

The result shown in the table (Table 4.10), shows ATWV of every selected query term and finally the average ATWV with 85%.In our experimental evaluation, the researchers understood that when the Ncorrect is less and Ntrue is higher the average ATWV will decrease and in reverse when the Ncorret and Ntrue values difference approaching to zero the average ATWV will increase which implies the performance of the system increase.

4.2.3.4. Performance Evaluation (Efficiency)

Running the system by using the environmental setup we mentioned in chapter 4 of this report, we found different efficiency results.

Time took for	**Test Speech**	**Duration of a speech**	**Time**
Segmentation	News	00:07:59	00:00:49(0 hours, zero-second and forty-nine second)
	Bible	00:08:31	00:01:09(0-hour, one-minute and nine second)

Figure 4. 11 Transcription and segmentation time by the STD

The result shown in the above table (Table 4.11), time taken to segment the given audio files (in hour, minute and second format) of news and bible speech which was approximately 8 minutes of speech. The decoding time is also shown in Figure 4.11 and Figure 4.12 which shows the time took to decode news and bible test speech respectively.

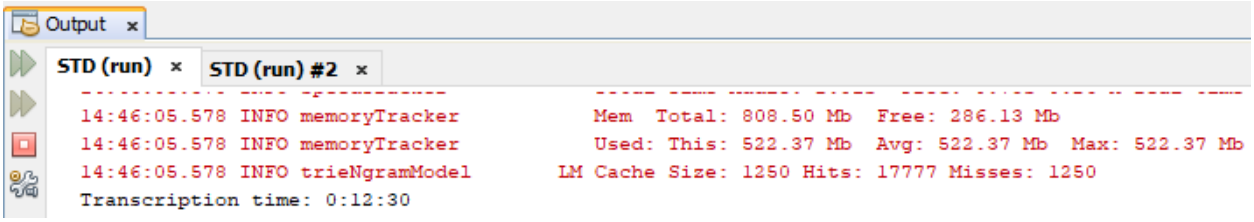

Table 4. 12 Time took to decode news speech

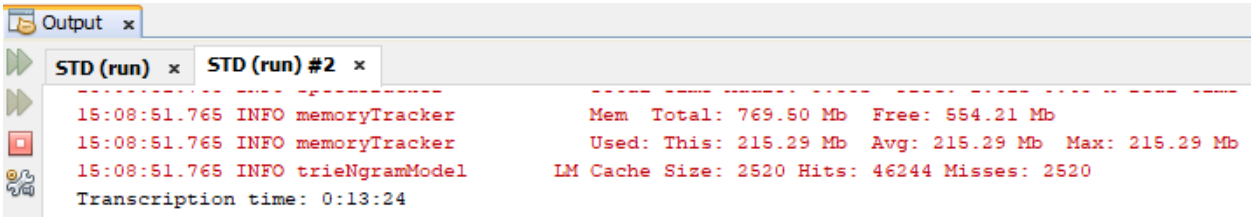

Figure 4. 12 Time took to decode Bible speech

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

This section outlines the conclusions drawn from the findings of the present study and possible future works in the field.

5.1. Conclusion

The improvement in technology and processing power of systems (computer), emergence and service improvement in social Medias create users to produce multimedia information easily. Accordingly, millions of audio files are generated and disseminated globally every day by individuals and organizations.

However, searching for the location of a particular word with its respective time interval is a challenge, particularly for languages like Amharic which are spoken by small number of the global population.

This study aimed to locate the time interval upon which the speech is located using user's query text and that was achieved by developing ASR. The ASR was developed first by comparing the automatically and manually segmented speech. For the automatically segmented speech study, we looked into phrase/word-like segmentation and compared it with sentence-like segmentation (in length of word). This helped us to select optimal segmentation to be applied for the development of text-based STD. We found that the performance of sentence-like segmentation was much more similar to that of the manually segmented test speech. Therefore, we used sentence-like segmentation for our automatic segmentation in our text-based STD. The acoustic model for the text-based STD was developed by using the speech corpus of LVCSR and tested with automatically segmented speech. First, we developed a LVCSR from the broadcast domain and tested it with an automatically segmented Bible and news domain. The result showed that using the same domain for both test and training speech corpus for ASR development performs better than using different domains. However we also go further to check the performance of ASR by combining the training speech corpus from the broadcast domain and the Bible speech by interpolating the LM's as well and using Bible as a test speech. We found that combining the training and LM's of two different domains showed better results than training ASR from one domain and test with speech from another domain. We used the acoustic model of the above

LVCSR from the news domain and by combining the LVCSR broadcast and Bible domain to develop STD. Therefore as the performance of ASR increases the possibility of recognizing the words by the STD system will increase. Finally, the developed STD was tested with different query words. There were words which retrieved and located perfectly. On the other hand, some words were not correctly recognized and located while we made our search. Therefore, to locate and get the exact location, we need to have the best ASR with low WER. If so we could locate query words with their respected time frame.

This research helps ASR development in such a way that it encourages researchers to use automatically segmented speech to test their ASR performance. Apart from this, the text-based STD helps users to search a particular spoken term easily with its time frame.

In our opinion, the text-based STD will help users to easily search time frame and in our study, we showed that it is possible to locate a time interval for a given spoken word from a given audio file.

5.2. Recommendation/Feature work

- The technique of comparing the automatically and manually segmented speech can be applied on the training speech corpus by using the same methodology and technique.
- Text-based STD can be extended to phrase and sentence level searching.
- Contextual searching could also be implemented apart from word and phrase searching.
- Since Amharic is morphologically reach language we recommend including steamers both on the query word and the transcribed text so that the effectiveness of the text-based STD will be improved.
- We recommend to extend this work to the health, legislative (court), and many other related domains.
- All search terms in this study are mostly nouns that require further study on user's preference query terms.
- Further study needs to be conducted to handle homophones using a given audio file and search term.
- The research can be extended into spontaneous speeches.
- Parameters used to segment the audio file was done using minimum silence and silence threshold which requires automatically estimating those parameters from the users audio file

- We recommend ways to minimize the running time (efficiency) of the text-based STD.
- A media player could be developed to mark the time frames, which will further enhance usability.

References

[1] J. Sallabank P. Austin, *The Cambridge Handbook of Endangered Languages*. New York, United States of America: Cambridge University Press, 2011.

[2] R.Nath,S.Kumar A.P.Singh, "A Survey: Speech Recognition Approaches and Techniques," in *2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering(UPCON)*, 2018.

[3] M.Larson G.J.F. Jones, *Spoken Content Retrieval: A Survey of Techniques and Technologies*.: Now Foundations and Trends, 2012.

[4] D.T. Toledano, P. Lopez-Otero, L. Docio-Fernandez, L. Serrano, I. Hernaez, A.Coucheiro-Limeres, J. Ferreiros, J.Olcoz , J. Llombart J. Tejedor, "Spanish, ALBAYZIN 2016 spoken term detection evaluation: an international open competitive evaluation in," *EURASIP Journal on Audio, Speech, and Music Processing*, 2017.

[5] D. T. Toledano,J.M. Ramirez ,A. R. Montalvo,J. I. Alvarez-Trejos J. Tejedor, "The Multi-Domain International Search on Speech 2020 ALBAYZIN Evaluation: Overview, Systems, Results, Discussion and Post-Evaluation Analyses," 2021.

[6] S. T. Abate, "Automatic Speech Recognition for Amharic," 2006.

[7] K. Kodlekere,K. Akshatha,S. J. Prasad A. Chaudhary, "Keyword Based Indexing of a Multimedia File," in *2017 IEEE International Symposium on Multimedia (ISM)*, Taichung, Taiwan, 2017, pp. 573-576.

[8] A. Hassen, "Develop an Audio Search Engine for Amharic Speech web Resources," AAU, Addis Ababa, Thesis 2019.

[9] A. Mohammed, "Video & Audio Content Search Engine (VACSE)," *Academic Journal of Science*, 2014.

[10] D.T. Toledano, P. Lopez-Otero, L.Docio-Fernandez, A.R. Montalvo, J.M. Ramirez, M.Penagarikano,L.J. Rodriguez-Fuentes J.Tejedor, "ALBAYZIN 2018 spoken term detection evaluation: a multi-domain international evaluation in Spanish," *EURASIP Journal on Audio, Speech, and Music Processing*, 2019.

[11] M.Batuhan, "Keyword Search for Low Resourced Language," Doctor of Philosophy 2017.

[12] M.Y.Tachbelie, W.Menzel S.T.Abate, "Amharic Speech Recognition: Past, Present and Future," in *Proceedings of the 16th International Conference of Ethiopian Studies, ed. by Svein Ege, Harald Aspen, Birhanu Teferra and Shiferaw Bekele, Trondheim 2009*, 2009, pp. 1391-1401.

[13] R. Mekonen, "Prosody Based Authomatic Speech Segmentation For Amharic," AAU, Addis Ababa, 2019.

[14] W.Menzel,B.Tafila S.T.Abate, "An Amharic speech corpus for large vocabulary continuous speech recognition.," , 2005, pp. 1601-1604.

[15] S.S.Khaing Y.W.Chit, "Myanmar Continuous Speech Recognition System Using Fuzzy Logic Classification in Speech Segmentation," in *2018 International Conference on Intelligent Information Technology*, 2018, pp. 14-17.

[16] A. R. Chowdhury, K. Fawaz, P. Ramanathan S.Ahmed, "Preech: A System for Privacy-Preserving Speech Transcription," in *USENIX Security Symposium (USENIX Security 2020)*, 2020, pp. 2703-2720.

[17] CMUSphinx. [Online]. https://cmusphinx.github.io/ retrived on August 31 2021

[18] S.Cass. IEEE Spectrum. [Online]. Top Programming Languages 2020,https://spectrum.ieee.org/at-work/tech-careers/top-programming-language-2020

[19] W.Menze S. T. Abate, "Syllable-Based Speech Recognition for Amharic".

[20] L.Besacier,M.Meshesha M.Woldeyohannis, "Amharic Speech Recognition for Speech Translation," in *Actes de la conférence conjointe JEP-TALN-RECITAL 2016*, 2016.

[21] K.Smaïli Y. Biadgline, "Parallel Corpora Preparation for English-Amharic Machine Translation," in *International work conference on artificial neural networks*, 2021.

[22] S.Eyassu, "Classifying Amharic News Text Using Self-Organizing Maps," 2005.

[23] B.Yimam, *የአማርኛሰዋሰው*. Addis Ababa,Ethiopia, 1986.

[24] W. Leslau, *Introductory grammar of Amharic*.: Wiesbaden : Harrassowitz, 2000.

[25] G.Amare, *Modern Amharic Grammar in a Simple Approach*. Addis Ababa, 1997.

[26] S.Teka, "Handling Pronunciation Variation Using Hybrid Approach in Continuous, Speaker Independent Speech Recognition for Amharic," 2014.

[27] A. Karibayeva,Z.h. Zhumanov U. Tukeyev, "Morphological segmentation method for Turkic language neural machine translation," 2020.

[28] Y.Assabie M.Abate, "Development of Amharic Morphological Analyzer Using Memory-Based Learning," in *International Conference on Natural Language Processing*, 2014.

[29] S.M. Abdou , S.E.Hamid, M. Rashwan A.E. Sakran, "A Review: Automatic Speech Segmentation," *International Journal of Computer Science and Mobile Computing*, vol. 6, pp. 308 – 315, 2017.

[30] D.G L.Mary, *Searching Speech Databases Features, Techniques and Evaluation*.: Springer Nature Switzerland AG, 2018.

[31] L.Parayitam,V.Appala M.Pala1, "Real-time transcription, keyword spotting, archival and retrieval for telugu TV news using ASR," *International Journal of Speech Technology*, pp. 435-439, 2019.

[32] G. J. F. Jones M. Larson, "A Survey of Techniques and Technologies.: Now Foundations and Trends," pp. 237-422, 2012.

[33] S.Kalantari, "Improving Spoken Term Detection Using Complementary Information," PhD Thesis 2015.

[34] D.G L.Mary, *Searching Speech Databases*.: Springer, Cham, 2018.

[35] P.Zhang,Y.Yan,J.Pan,X. Na X.Wang, "Handling OOVWords in Mandarin Spoken Term Detection with an Hierarchical n-Gram Language Model," *Chinese Journal of Electronics*, vol. 26, pp. 1240-1244, 2017.

[36] D. Karakos,K.Narasimhan,M. Davel C.V. Heerden, "Constructing sub-word units for spoken term detection," in *ICASSP 2017 - 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[37] A. Mousa,H. Kuo,H.Soltau L.Mangu, "Morpheme-based feature-rich language models using Deep Neural Networks for LVCSR of Egyptian Arabic," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference*, 2013, pp. 8435-8439.

[38] S.Young M. Gales, "The Application of Hidden Markov Models in Speech Recognition," vol. Vol. 1, pp. 197-304, 2007.

[39] M. Cooke, P. D. Green J. Barker, "Robust ASR based on clean speechmodels: an evaluation of missing data techniques for connected digit recognitionin noise," in *in Proceedings of Eurospeech*, Aalberg, Denmark, 2001, pp. 213-216.

[40] A. Dennai,Y. Elmir S. Benk, "A Study on Automatic Speech Recognition," *Journal of Information Technology Review*, vol. 10, 2019.

[41] B.i W.Gawali,Pravin. Yannawar S. K.Gaikwad, "A Review on Speech Recognition Technique," *International Journal of Computer Applications*, vol. 10, 2010.

[42] P.P. Shrishrimal,R.R.Deshmukh S.K.Saksamudre, "A Review on Different Approaches for Speech

Recognition System," *International Journal of Computer Applications*, vol. 115, 2015.

[43] Vi.Mago M.Qudar, "A Survey on Language Models," 2020.

[44] M. Y.Tachbelie, "Morphology-Based Language Modeling for Amharic," University of Hamburg, Desertation 2010.

[45] D.Kiela,H.Schwenk,L.Barrault A.Conneau, "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[46] J.Goodman S.F. Chen, "An empirical study of smoothing techniques for language modeling," *Stanley F. Chen and Joshua Goodman*, 1998.

[47] Hamda M. M. Eljagmani, "Arabic Speech Recognition Systems," Florida Institute of Technology, Melbourne, Florida, Thesis 2017.

[48] P.Kwok, E.Gouvêa, B.Raj,R.Singh, W.Walker, M. Warmuth, P.Wolf P.Lamere, "THE CMU SPHINX-4 SPEECH RECOGNITION SYSTEM," 2003.

[49] C.Liu, R. A. Calvo, K.McCabe,S. C. R. Taylor, B. W. Schuller, K.Wu J.Y. Kim, "A Comparison of Online Automatic Speech Recognition Systems and the Nonverbal Responses to Unintelligible Speech," 2019.

[50] M. Kamran Malik,K. Mehmood,I.Makhdoom, M. Malik, "Automatic speech recognition: a survey," November 2020.

[51] N.Geri V.Silber-Varod, "Can automatic speech recognition be satisficing for audio/video search? Keyword-focused analysis of Hebrew automatic and manual transcription," *Online Journal of Applied Knowledge Management*, vol. 2, pp. 104-121, 2014.

[52] J. Neto H.Meinedo, "Automatic Speech Annotation and Transcription in a Broadcast News task," 2003.

[53] A.Rajpoot P. Sharma, "Automatic Identification of Silence,Unvoiced and Voiced Chunks in Speech," *Journal of Computer Science & Information*, pp. 88-96, 2013.

[54] A. Kinghorn M. Greenwood, "SUVING: Automatic Silece/Unvoiced/Voiced Classification of Speech," 1999.

[55] I.Ayass,M. Ghareeb,Z.El-Bazzal,M.Raad M. N. Al Laham, "Audio Indexing for YouTube," in *2015 Fifth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, Beirut, Lebanon, 2015.

[56] Dogan. Can,M. Saraclar S.Parlak, "Turkish Broadcast News Transcription and Retrieval," 2009.

[57] J.R.Glass, "Statistical trajectory models for phonetic recognition.," , 1994.

[58] ethio dawit tube. [Online]. https://www.youtube.com/channel/UCVVlFOCbsXPKsa-Xsya8xgQ,Retrived on Feb 10,2021

[59] WordProject. [Online]. https://www.wordproject.org/bibles/index.htm retrived sep 1 2021

[60] M. Gasser,T.Takara T.Anberbir, "Grapheme-to-Phoneme Conversion for Amharic Text-to-Speech System," pp. 68-73, 2011.

[61] [Online]. https://www.bible.com/bible/1260/MAT.10.NASV retrived on August 2021

[62] W.Menzel S.Tefera, "An Amharic speech corpus for large vocabulary continuous speech recognition.," , 2005, p. 5.

[63] G.Gosztolya, "On the Concept of Correct Hits in Spoken Term Detection," in *Conference: Artificial Intelligence and Cognition (AIC)*, 2014, pp. 75-171.

[64] L.Toth G. Gosztolya, "Spoken term detection based on the most probable phoneme sequence," in *9th IEEE International Symposium on Applied Machine Intelligence and Informatics*, 2011, pp. 101-106.

[65] A.Stolcke, "an extensible language modeling toolkit," in *In Proceedings of International Conference on Spoken Language Processing*, 2002.

[66] A.Q.Syadida,D.R.I.M.Setiadi,A.Setyono M. Muljono, "Sphinx4 for Indonesian continuous speech recognition system," in *2017 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2017, pp. 264-267.

[67] T. Anastasakos, H. Jin, L. Nguyen, R. Schwartz F. Kubala, "Transcribing radio news," , 1996, pp. 598–601.

[68] M. Kleber, C. Kao, O. Kimball, T. Colthurst, S.A D.R.H. Miller, "Rapid and accurate spoken term detection," in *in Proc. Interspeech*, 2007, pp. 314-317.

[69] S.Young M.J.F. Gales, "The Application of Hidden Markov Models in Speech Recognition," *Foundations and Trends® in Signal Processing*, vol. 1, pp. 197-304, 2007.

[70] L.Parayitam,V.Appala M.Pala, "Real-time transcription, keyword spotting, archival and retrieval for telugu TV news using ASR," *International Journal of Speech Technology*, pp. 435-439, 2019.

[71] D. T. Toledano, P.Lopez-Otero, L. Docio-Fernandez J. Tejedor1, "ALBAYZIN 2018 spoken term detection evaluation: a multi-domain international evaluation in Spanish," p. 37, Sep. 2019.

[72] K. Kodlekere,K. Akshatha,S. J. Prasad A. Chaudhary, "Keyword Based Indexing of a Multimedia File," in *2017 IEEE International Symposium on Multimedia (ISM)*, Taichung, Taiwan, 2017, p. 4.

Appendix

Appendix A: Phonetic Dictionary

ኢየሱስም	hh i j aa s u s ee m ee
በይሁዳ	b aa j ee h u d a
ቤተ	b e t aa
ልሄም	l ee h e m ee
በንጉሱ	b aa n ee g u s u
በሄሮድስ	b aa h e r o d ee s ee
ዘመን	z aa m aa n ee
በተወለደ	b aa t aa w aa l aa d aa
ጊዜ	g i z e
እነሆ	hh ee n aa h o
ሰበአ	s aa b ee hh a
ሰገል	s aa g aa l ee
የተወለደው	j aa t aa w aa l aa d aa w ee
የአይሁድ	j aa hh a j ee h u d ee

Appendix B: Sample Training Transcription

<s> ዘካርያስም መልአኩን እኔ ሽማግሌ ነኝ ምስቴም በእድሜዋ እርጅታለችና ይህን በምን አውቃለሁ አለው </s> (lukas14)

<s> መልአኩም መልሶ እኔ በእግዚአብሄር ፊት የምቆመው ገብርኤል ነኝ እንድናገርህም ይህችንም የምስራች እንድሰብክልህ ተልኬ ነበር </s> (lukas15)

<s> እነሆም በጊዜው የሚፈጸመውን ቃሌን ስላላመንህ ይህ ነገር እስከ ሚሆን ቀን ድረስ ዲዳ ትሆናለህ መናገርም አትችልም አለው </s> (lukas16)

<s> ህዝቡም ዘካርያስን ይጠብቁት ነበርበቤተ መቅደስም ውስጥ ስለ ዘገየ ይደነቁ ነበር </s> (lukas17)

<s> በወጣም ጊዜ ሊነግራቸው አልቻለም በቤተ መቅደስም ራእይ እንዳየ አስተዋሉእርሱም ይጠቅሳቸው ነበርድዳም ሆኖ ኖረ </s> (lukas18)

<s> የማገልገሉም ወራት ሲፈጸም ወደ ቤቱ ሄደ </s> (lukas19)

Appendix C: Sample Test Transcription

<s> የኢንተርኔት አገልግሎትንም በተመለከተ በክልሎች በዞኖችና በአዲስ አበባ በተለያዩ ቦታዎች የአገልግሎት ማእከሎችን ለማቋቋም መታቀዱን አብራርተዋል </s> (01_d501021)

<s> በዚህ ም አነስተኛ መስኖዎችን በማስፋፋት አገሪቱን በተደጋጋሚ ከሚያጠቃት ድርቅና የምግብ እህል እጥረት ለማቃለል እንደሚቻል ጠቁመዋል </s> (01_d501022)

<s> ይህ እንዳይሆን በህይወታቸው መስዋእትነት ያስገኙትን የኢትዮጵያዊነት ክብር እንግበን ለቅድስት ሀገራችን አንድነትና ሉአላዊነት ሰላምና ብልጽግና በሁብረት እንቁም </s> (01_d501023)

<s> የመጀመሪያው ነጥብ በተቃዋሚ ሀይሎች ሰፊር በጋራ አቋም ላይ በጋራ ለማቆም ያየነው ጽናት አነስተኛ መሆኑ ነው </s> (01_d501024)

<s> አቅም በሚፈቅደው መሰረት እስቀድሞ መዘጋጀት ስለሚፈልጉ የሚያጠራጥር እይሆንም ስትል የአሜሪካ ድምጽ ዜና ዘጋቢ ገልጻለች </s> (01_d501025)

<s> ከለቀስተኞቹ መካከል አብዛኛዎቹ ጸሀዩ ዛሬ ተገለጠ ሀይለስላሴም ሲ ማሩን አለማችን አበታችን የእፍሪካ አቦት የአለም አቦት በማለት ነበር ሀዘናቸውን የሚገልጹ </s> (01_d501026)

<s> ከዚህ ጋር ለሱዛለዊ እገልግሎት መሰለፏም ውጤታማ መሆኗም እድናቆትን አትርፎላታል </s> (01_d501027)

<s> የኮሚሽኑ ውሳኔና የቢተ ከህኑ ተቃውሞ </s> (01_d501028)

<s> ይህንንም ባድመንና ሽራሮን በመውረር እውን እርጓል </s> (01_d501029)

<s> ጋዜጠኞችን መለያየታቸው ብዙዎችን አሳዝኗል </s> (01_d501030)

<s> ባለፈው ሰኞ እለት ደግሞ በቲሌቭቭዥን ሌላ ሽልማት ሲሸለም ተመልከትኩ </s> (01_d501031)

Appendix D: Sample of Most and Less Frequent News Words

Decoded words by recognizer	Hit words	Frequency
Segment00.wav በዚህ ቃል አገልግሎት በተመለከተ በክልሎች በዚህ ሙስና አዲስ አበባ እለት ያሉት ውጭ ያሉት ሀገራችን ለማቋቋም ከተማ ጋር ቃል	አገልግሎት	36
	በተመለከተ	13
	በክልሎች	1
	አዲስ	235
	አበባ	113
	ለማቋቋም	9
Segment01.wav በዚህ ላይ ይገኛሉ ሰው ተብሎ ጽፊት አንድነት በጋራ ልማት ጋር ከተማ የምግብ እጥረት አራት ቀን የሚችል ጠቁመዋል	የምግብ	9
	እጥረት	4
	ጠቁመዋል	19
Segment02.wav ሌላ ማለታቸውን ቃል ነገር ግን ኤርትራ ለመከላከል ቀን ግን ሀገራችን አንድነትና ና ላይ ነው ሰላምና ግን ህግና በህብረት ነበር	ሀገራችን	38
	ሰላምና	16
	በህብረት	1
segment03.wav ይመስላል ይጠበቃል መለስ ጋር በጋራ አቋም ላይ በጋራ መልኩ አለም ጽናት አነስተኛ ሆነው አይደለም ሊሰጠው መሰረት ከስ ደግሞ መዘጋጀት የሚገኙ	በጋራ	7
	አቋም	20
	ላይ	547
	በጋራ	7
	ጽናት	7
	አነስተኛ	10
	መዘጋጀት	2
	የሚገኙ	120
segment4.wav ቀይ ጋዜጠኞች መካከል አብዛኛዎቹ ዛሬ ዛሬ ተገለጠ	መካከል	65
	አብዛኛዎቹ	16
	ዛሬ	194
	ተገለጠ	32
segment05.wav ከዚህ ጋር የስራ ላይ ያሉት መሰረት ሁሉ ብቻ መሆኗም	ከዚህ	43
	ጋር	163
	መሆኗም	1
segment06.wav ከዚህ ጦርነት ሁለት አለ		
segment07.wav የኮሚሽኑ ዛሬ ና ነገር ግን ተቃውሞ	የኮሚሽኑ	3

	ተቃውሞ	67
segment08.wav ችለናል ባድመንና ሽራሮን በመውረር እውን ቃል	ባድመንና	2
	ሽራሮን	3
	በመውረር	1
	እውን	8
segment09.wav ታሪክ መቶ ና ላይ ናቸው ብዙዎችን ካሳ ዘመን	ብዙዎችን	1
segment10.wav ባለፈው ሰኞ የቀሩት አዲስ ሌላ ሽልማት ሲሸለም በማለት	ባለፈው	59
	ሰኞ	22
	ሌላ	110
	ሽልማት	25
	ሲሸለም	0
segment11.wav ተቃውሞ ልጅ ካሳ ስምንት ሮናልዶ ላይ ለማወቅ ተችሏል	ስምንት	60
	ለማወቅ	53
	ተችሏል	44
segment12.wav ሌሎቹ በሙሉ ድምጽ ናቸው	ሌሎቹ	13
	በሙሉ	48
segment13.wav ይህ ማህበር የስራ ዘመን በኋላ ግን መልኩ ነው		
segment14.wav በኋላ ላይ ናቸው ያሉት አራት ሁለት ነገር ደግሞ በሰላም ገልጿል ሲል አለም ሙከራ ነበር ጽፈት ቤት መግለጫ አስታውቋል	በሰላም	4
	ሲል	23
	ቤት	190
	መግለጫ	43
	አስታውቋል	36
segment15.wav ኳሶች ጋር የነበሩ በአዲስ ችለናል እጃቸውን ተችሏል የፖሊስ መግለጫው ነው ገልጿል	እጃቸውን	10
	የፖሊስ	10
	መግለጫው	11
	ገልጿል	92
segment16.wav ተብሎ ቡድኑ ማስወገድ ነበረባቸው	ማስወገድ	12
	ነበረባቸው	1
segment17.wav ክብር ላይ አስታውቋል ቃል ለማቋቋም አለም ሰው አቶ ና ምንጮች ከሰጡት ድጋፍ ጊዜው	ከሰጡት	1
	ድጋፍ	21
segment18.wav ይህ ሁሉ የስራ ቃል		
segment19.wav በዚህ ሌላ አጋጣሚ ግጭት ኤርትራ ተጠቃሚ ሚኒስትሩ በተለይ አስመራ ላይ ያሉት ግን ሰፊ እንደሆነ ገልጿል	በዚህ	95
	በተለይ	50
	ሰፊ	10
	እንደሆነ	83
segment20.wav ድፍረት ባይሆንበኝ ስህተት ነው	ድፍረት	4
	ባይሆንበኝ	1
	ስህተት	31
	ነው	
segment21.wav ቃል አቶ ቀን ዘመን ታሪክ አራት ክልል አስተዳደር መቶ ስምንት ይቻላል	ዘመን	12
	ታሪክ	52
	አራት	99
	ይቻላል	21
segment22.wav ዲሞክራሲ መልእክት ቅነሳ ግን ጋር የበለጠ ምን ለማለት የዋጋ ለውጥ ነገር ግን አላደረገም	ቅነሳ	4
	ግን	428
	የዋጋ	2
	ለውጥ	19
	አላደረገም	4
segment23.wav በኋላ ሌላ ብቻ ነው ለማለት		
segment24.wav ቃል ለመረዳት ተችሏል ተብሏል ተጨማሪ ጦርነት አለ ላይ	ጦርነት	36

በወያኔ ቃል ውስጥ ነው		
segment25.wav ሁሴን አይዲድ አንድነት አቋም ሁሉ ሰላምና ልማት ነው	ሁሴን	18
	አይዲድ	17
	አቋም	20
segment26.wav በዚህ ተስማሙ ለማለት ለማን ለማን ከዚህ ካሳ ላይ የለም ሲል የጦር ሰፈር ሲል ሀሳብ ተስማሙ	በዚህ	95
	ለማን	22
	ለማን	22
	ካሳ	11
	ሀሳብ	54
segment27.wav ሮናልዶ መናገር ገልጸዋል	ሮናልዶ	11
	መናገር	9
	በማለት	91
segment28.wav ታሪክ ጋር ክስ ሀይል ውስጥ መልእክት አለም ውስጥ ሌላ አቅም ሁሉ አለም ላይ መግለጫ አስታውቋል በማለት መልሷል	መልሷል	7
	አለም	65
segment29.wav ከዚህ ጋር ክስ መግለጫ ለመከላከል ተገናኝቶ ያሉት ናቸው ኳሶች መስቀል ናቸው	ተገናኝቶ	1
	ናቸው	720
	ኳሶች	4
segment30.wav ከተማ የሚገኙ ቦታ ድርጅት አለ ያሉት ግን እንደ መግለጫው		
segment31.wav ተችሏል መገልገያ በጸረ ሙስና ኮሚሽን ክስ ላይ በቀጥታ ፕሮግራም ጦርነት ሀይሎች ውጪ ከኢትዮጵያ ጋር ራሳቸውን ሀገራችን ተጠቁሟል	በጸረ	4
	ሙስና	37
	ኮሚሽን	61
	ክስ	143
	ላይ	547
	ውጪ	62
	ተጠቁሟል	16
segment32.wav ታሪክ አገሮች መካከል ለምን ቦታ ላይ ነው		
segment33.wav ይህ ሁሉ ልጅ ይህ ማህበር ኤርትራ ሰራዊቷን ክስ አቅም ጋር	ማህበር	21
	ኤርትራ	606
	ሰራዊቷን	5
segment34.wav ቀይ ራሴ የት ቦታ ሆኜ ስራዬን ባለስልጣናት ማለታቸውን እኔ ጊዜ ቀርቶ ባለፈው ተጨማሪ ቃል ቦታ ላይ አስደምጧል	ራሴ	4
	የት	109
	ቦታ	78
	ሆኜ	13
	ስራዬን	2
	ማለታቸውን	4
	አስደምጧል	1
segment35.wav ይህ ጋር በማለት		
segment36.wav ቃል መለስ የተባለው የጦር ጀኔራሎች ስብሰባ እኔ አሁን መድረክ ነው ተብሏል	መለስ	131
	የጦር	33
	ጀኔራሎች	8
	ስብሰባ	112
	ተብሏል	23
segment37.wav ቃል ነው ልጅ አቶ አክሎ ገልጿል		
segment38.wav ታሪክ ሀላፊ የሚገኝበት ከተማ ነው	የሚገኝበት	1
	ከተማ	112
	ነው	
	ሆነው	92
	ይነገራል	7

segment39.wav ፋና ክስ አቶ የሚላቸው ስምንት መግለጫ ሌሎች ሌላ ሆነው ቃል እውን የሚል ላይ ገልጿል ሁለቱ ነው ይነገራል	ስምንት/sementu	60
segment40.wav ታሪክ ደግሞ ዘመን ጦር ተሰማ የሚች ምንጮች ሆነው		
segment41.wav ቃል በር ላይ ቀይ በኋላ ኤርትራ ላይ ኢሳያስ ባለስልጣናት ሲል ላይ ከሚል አቋም መቶ በተገኙበት ገልጿል	ገልጿል/getltsewal	92
segment42.wav ጊዜ ውሀ ለማቋቋም ከሚል በሚገኘው መኖሪያ ቤቱ	ውሀ	28
	በሚገኘው	15
	መኖሪያ	8
	ቤቱ	37
segment43.wav ድጋፍ ነበር ሰው ሁሉ የደረሰበት አቅም	የደረሰበት	2
segment44.wav ግን ለማስተላለፍ አለም በኋላ በኋላ ነው የት አለ	በኋላ	52
	በኋላ	
	ነው	
segment45.wav ፖሊስ ጋር በጋራ ተወስደው መታሰራቸው ቀርቶ ቀርቶ ቃል እኔ ራሴ ለበርካታ ጊዜ ድረስ ለማለት ነው ቃል ነው ሙሉ ቃል ውጭ ገልጸዋል	ጊዜ	112
	ድረስ	66
	ገልጸዋል	141
segment46.wav ቃል ሁሉ ሌሎች ሌሎቹ አለም ማለታቸውን ጋር መናገር ለመከላከል አይነት ናቸው ህግና ደንብ መዘጋጀት ይገባል አራት ና መግለጫ በግልጽ ላይ	ሌሎች	146
	ህግና	15
	ደንብ	19
	መዘጋጀት	2
	ይገባል/endemigeba	32
segment47.wav ከዚህ ደግሞ ራሳቸውን ይቻላል በመቶ እለት ለማቋቋም ቀን ብለዋል በማለት አስራ አንድ ፍጻሜ ሲል ጦር በኋላ ይህ ጊዜ የስፖርት ጋዜጠኞች እየጻፉ ነው አሁን ላይ ግን ሁለት ሰው ሁሉ	ራሳቸውን	34
	የስፖርት	7
	ጋዜጠኞች	36
	እየጻፉ	1
	አሁን/ahunem	1
segment48.wav የአየር ሀይል በውሳኔው ተጠቃሚ ሳይሆን መስቀል ከዜና ምንጮቻችን ለመረዳት ችለናል ምን	ሀይል	58
	ተጠቃሚ	5
	ሳይሆን	36
	ምንጮቻችን	21
	ለመረዳት	8
	ችለናል	11
segment49.wav ተቃውሞ ወደ ሌላ ጽድቁ		
segment50.wav ቀይ ለመከላከል ሂዱ ከዚህ ጋር አለ ቅዳሜ ፕሬዝዳንት በጸረ መንግስት አራት ሀላፊ ኤርትራ ሁሉ	ሂዱ	5
	ከዚህ	43
	ቅዳሜ	26
	ፕሬዝዳንት	64
segment51.wav ፋና ዲሞክራሲ ሰኞ ሙላት አፍሪካ በተለይ ተጠቃሚ ድርጅት በይፋ ተዘግቶ ሰራተኞች መበተናቸው ታወቀ	ፋና	9
	ዲሞክራሲ	33
	ድርጅት	59
	በይፋ	7
	ተዘግቶ	4
	ሰራተኞች	38
	መበተናቸው	2
	ታወቀ	51
segment52.wav ይህ ዛሬ ላይ ያሉት ቃል መቶ ሜትር አራት ሜትር ሜዳ ሙሉ ቃል የኳስ ምንጮች ሜዳ ይሆናል	መቶ	166
	ሜትር	6
	የኳስ	2
	ሜዳ	74

	ይሆናል/saied yahelal	17
segment53.wav ተብሎ ሌላ ሌላ የኢነጋማ እንደ አቶ ክፍሌ ሙላት ግን አገሪቱን ወደ ክልል ነው ያሉት ሁሉ	የኢነጋማ	3
	አቶ	167
	ክፍሌ	2
	ሙላት	3
	ነው	1671
	ያሉት	
segment54.wav የኬንያው ፕሬዝዳንት ጋር በቀጥታ ጋር በኬንያና ሊግ አለም ጋር አካባቢ በተነሳው ግጭት መልኩ አራት ምንጮች ጠቁመዋል አገልግሎት ግን ራሱን ብቻ ሂደትና መልክ አቋም የሚችል የሚገኙ የድርጅቱ ነው	የኬንያው	10
	ፕሬዝዳንት	64
	በኬንያና	3
	ጋር	163
	በተነሳው	2
	ግጭት	18
	ምንጮች	60
	ጠቁመዋል	19
	ራሱን	8
	ሂደትና	0
	የሚችል	9
	ነው	1671
segment55.wav ይጠበቃል ቀይ መስቀል ፊት ለፊት አበባ በሙቶ ሌላ ሰው ይቻላል ለማባረር ይገባል	ቀይ	16
	መስቀል	10
	ለማባረር	1
segment56.wav ይህ ፕሮግራም በማለት		
segment57.wav የሚች ቤት በር ላይ ተወስደው ሊግ በተደጋጋሚ	የሚች	1
	የላቸውም/yalachew	6
	ያደርጋሉ	8
segment58.wav የላቸውም ያደርጋሉ ከሚል ንቀት እድል የጀርመን በኋላ ይናገራሉ	ገልጿል/geltewal	92
segment59.wav የሚል ሌላ ቦታ ነው መለስ አስቀድሞ ገልጿል		
segment60.wav ትኩረት ሊሰጠው ይገባል ገልጿል አስራ ላይ በቀጥታ የሚመለከታቸውን ሁሉ በኋላ ይህ ተጨማሪ መልእክት ለማስተላለፍ እድል ልጅ ተስፋ ያደርጋሉ	ትኩረት	31
	ሊሰጠው	3
	ይገባል/endemigeaba	32
	በቀጥታ	4
	የሚመለከታቸ ውን	0
	ሁሉ	94
	ተጨማሪ	101
	መልእክት	15
	ለማስተላለፍ	1
	ተስፋ	51
segment61.wav ከሚል ሌላ ደግሞ የት የት ነው	ነው	1671
segment62.wav ቀርቶ ገልጿል ከአንድ ሽልማት መቶ ዘጠና ነበር	ዘጠና	22
	ነበር	--
segment63.wav ቃል ሲል ዘጋቢ ውጪ የሚያበረታታ ነው	የሚያበረታታ	1
	ነው	1671

Appendix E: Sample of Most and Less Frequent Bible Words

Decoded words by recognizer	Hit words	Frequency in training corpus
segment00.wav ድረስ ሰው በይሁዳ ቤተ ልሄም በግብጽ በሄሮድስ ዘመን ደግሞ ልጇ ጊዜ	በይሁዳ	4
	ቤተ	36
	ልሄም	2
	በሄሮድስ	1
	ዘመን	19
	ጊዜ	117
segment01.wav ጀምሮ ፈራ ሰገል	ሰገል	--
segment02.wav ሁሉ ድምጽ ራቅ አይተን ልንሰግድለት መጥተናልና ያለ	አይተን	--
	ልንሰግድለት	--
	መጥተናልና	--
segment03.wav የህጻኑን ሁሉ	ሁሉ	188
segment04.wav ክርስቶስ ደግሞ ብሎ ልጅ ጠየቃቸው	ክርስቶስ	8
	ጠየቃቸው	1
segment05.wav ከይሁዳ ገዢዎች ከአባታቸው	ከይሁዳ	2
	ገዢዎች	2
segment06.wav ህዝቤን ሰው ሁሉ የሚጠብቅ ራስህን ቃል ሰው ቃል ነው	ህዝቤን	1
	የሚጠብቅ	1
	ነው	1517
segment07.wav ከዚህ ላይ ስለ ጋር ግን እርሱ ኮከቡ	ከዚህ	48
segment08.wav ከጌታ ልጅ ደግሞ ዘመን		
segment09.wav ከእነርሱ በጥንቃቄ ቀርበው	ከእነርሱ	11
	በጥንቃቄ	3
	ቀርበው	3
segment10.wav ከጌታ ልሄምም		
segment11.wav ከእነርሱ	ከእነርሱ	11
segment12.wav እርሱም ሰምተው ሄደ	እርሱም	48
	ሰምተው	6
segment13.wav ቃል ነው በምስራቅ ይልቅ ኮከብ	በምስራቅ	--
	ኮከብ	1
segment14.wav ራሄል ተቀመጥ ጽድቅን ተሰማ ከቶ ነበር	ነበር	244
segment15.wav ሁሉ ከቶ ነው ቃል ጊዜ በታላቅ ደስታ እጅግ ደስ አላቸው	በታላቅ	1
	ደስታ	8
	እጅግ	21
	ደስ	21
	አላቸው	70
segment16.wav ከእናቱ ከማርያም ጋር ሂድ	ከእናቱ	--
	ከማርያም	1
	ጋር	208
segment17.wav ቃል ሰገዱለት	ሰገዱለት	--
segment18.wav ኢየሱስ እንዳይመለሱ በል ከፍተው	እንዳይመለሱ	--
segment19.wav ኢየሱስ		
segment20.wav አንተ ግን ይፈልገዋልና ተነሳ	ይፈልገዋልና	--
	ተነሳ	2
segment21.wav ስለ ምእራፍ	ተነስቶ	9

segment22.wav ፈጽሞ ተነስቶ ህጻናት እናቱን ፍሬ ከተማ ከጌታ ዘንድ	ህጻናት	20
	እናቱን	2
	ከጌታ	1
	ዘንድ	62
segment23.wav ኢየሱስም ብሎ ድረስ ደስ አለው	ድረስ	68
segment24.wav ከዚያ ተራራ ሄሮድስ ጫፍ ይልቅ ታላቅ በግብጽ ጊዜ	ከዚያ	15
	ሄሮድስ	3
	ጊዜ	117
segment25.wav ያን ጊዜ በገሊላ ድረስ ድምጽ በራማ ተሰማ ሁለት ነው	ያን	12
	ድምጽ	20
	በራማ	1
	ተሰማ	5
segment26.wav ኢየሱስም ከሞተ በኋላ ጀምሮ የጌታ መላእክት	ከሞተ	--
	ከሞተ	--
	በኋላ	66
	የጌታ	5
	መላእክት	8
segment27.wav ይህ ሰው ነፍስ አለቆች ሞተዋልና ተነሳ ሁሉ	ነፍስ	5
	ተነሳ	2
segment28.wav ደስ ተነስቶ ህጻኑንና ማዶ ያለ ነው	ተነስቶ	9
	ህጻኑንና	--
segment29.wav ድንጋዮች ወደ ገሊላ ጋር ሂድ	ወደ	470
	ገሊላ	6
	ሂድ	6
segment30.wav ጨለማ ድረስ እንደ ማር እርሱ		
segment31.wav ከዚያ ወራት መጥምቁ የሀንስ	ወራት	25
	መጥምቁ	3
	የሀንስ	13
segment32.wav በንጉሱ ሰማያት ቀርባለችና ንስሀ ግቡ	ሰማያት	--
	ቀርባለችና	--
	ንስሀ	9
	ግቡ	1
segment33.wav ራሱም ወንዝ የግመል ጠጉር ልብስ ነበር	ራሱም	1
	የግመል	--
	ጠጉር	1
	ልብስ	9
	ነበር	244
segment34.wav ኮከቡንም ጠፍር ይታጠቅ ነበር	ኮከቡንም	--
	ጠፍር	1
	ይታጠቅ	--
	ነበር	244
segment35.wav አቁሞ ከተማ የበረሀ ማር ነበር	የበረሀ	--
	ማር	--
	ነበር	244
segment36.wav ዳሩ ግን ከፈሪሳውያንና ከሰዱቃውያን	ዳሩ	5
	ግን	402
	ከፈሪሳውያንና	--
	ከሰዱቃውያን	--
segment37.wav ቀንና ልጇ ሁለት ልጆች ከሚመጣው ቀጣ ግን እርሱ ማን መንገድ እጅ	ልጆች	31
	ከሚመጣው	--
	ቀጣ	3
	ማን	58
segment38.wav ጊዜ ለንስሀ የሚገባ ፍሬ አድርጉ	ለንስሀ	1

	የሚገባ	6
	ፍሬ	10
	አድርጉ	5
segment39.wav ዳሩ		
segment40.wav ጋር ቃል ሁሉ		
segment41.wav ተነስቶ ነው ተራራ ምድር ኢየሱስ ኣመት		
segment42.wav ከእርሱ ሰው በዛፎች ስር በታንኳ	በዛፎች	1
	ስር	--
segment43.wav ልጇ መልካም ፍሬ ጊዜ ማዶ ዛፍ ሁሉ ይቁረጣል እሳትም ይጣላል	መልካም	19
	ፍሬ	10
	ዛፍ	4
	ይቁረጣል	--
	እሳትም	1
	ይጣላል	1
segment44.wav ድምጽ ለንስሀ ሁሉ	ለንስሀ	1
segment45.wav ጫማውን ሸሸ ቃል ዘንድ የማይገባኝ ጊዜ በኋላ የሚመጣው ግን	ጫማውን	--
	ዘንድ	62
	በኋላ	66
	የሚመጣው	3
segment46.wav ስለ የሚገባ አባት		
segment47.wav ደስ ወደ መጡ ከቶ		
segment48.wav እነርሱም ደግሞ ነው	ነው	1517
segment49.wav ያን ጊዜ ኢየሱስ በኋላ ስለ ከሞተ ገሊላ ወደ ዮርዳኖስ	ኢየሱስ	19
	ገሊላ	6
	ወደ	470
	ዮርዳኖስ	1
segment50.wav የሀንስ ግን	የሀንስ	13
segment51.wav ጌታን ከተጠመቀ ስለ ልጇ ማን	ከተጠመቀ/Letemek	--
segment52.wav ይህ ዘመን ሰው		
segment53.wav ይህን ከተጠመቀ በኋላ ግን	ከተጠመቀ	--
	በኋላ	66
segment54.wav ፍሬ ነው		
segment55.wav ሲወርድ		
segment56.wav ጀምሮ ንጉስ ተሰማ ጠጉር		
segment57.wav ቃል ስለ ልጇ ምእራፍ አራት	ምእራፍ	19
	አራት	83
segment58.wav ጊዜ ወዲያው	ወዲያው/wediya	12
segment59.wav ታላቅ ነውና ገባ ሌሊትም ከቶ በኋላ ታላቅ	ሌሊትም	--
	በኋላ	66
segment60.wav ቃል መጡ		
segment61.wav ድምጽ እንዲህ ጋር ልጇ እንጀራ ልጅ ሁሉ ይባላል	እንዲህ/enezihen	92
	እንጀራ	9
segment62.wav ድረስ መልሶ	መልሶ	32
segment63.wav ከዚህ ወራት ኢየሱስ ብቻ ቅዱስ ከተማ ወሰደው ነው	ከዚህ	48
	ቅዱስ	22
	ከተማ	
	ወሰደው/wesedewe	1

	na	
segment64.wav ደስ በመቅደስ ጫፍ ላይ ሁሉ	በመቅደስ	1
	ጫፍ	3
	ላይ	419
segment65.wav አባት ድረስ ማን ቃል ነው		
segment66.wav ደግሞ አለን		
segment67.wav በዳ ታች ራስህን ሁሉ አለ	ራስህን	2
	አለ/alew	89
segment68.wav ይፈጸም		
segment69.wav ጌታን ላይ አትፈታተነው ተብሎ ደግሞ ደስታ ላይ	ጌታን/geta	3
	አትፈታተነው	1
	ተብሎ	52
	ደግሞ	274
segment70.wav ከዲያብሎስ እጅግ ደግሞ ሆነ ተራራ ወሰደው	ከዲያብሎስ/diabilos	1
	እጅግ	21
	ሆነ	84
	ተራራ	3
	ወሰደው	1
segment71.wav ያለ ማን መንግስታት ሁሉ ከውሀ ጨለማ ስር	ያለማን/yalemenem	--
	መንግስታት	44
	ሁሉ	188
segment72.wav ዳሩ ጊዜ ኢየሱስ	ጊዜ	117
segment73.wav ጌታን ከዚህ ቃል		
segment74.wav ያን ጊዜ ጀምሮ ሰው	ያን	--
	ጊዜ	117
segment75.wav ጊዜ ሶስት	ጊዜ	117
segment76.wav ጥላ ይህን ትቶ በዛብሎንና በንፍታሌም ጋር	ትቶ	3
	በዛብሎንና	--
	በንፍታሌም	--
segment77.wav ገባ አጠገብ ቃል እንደ ቅፍርናሆም መጡ	አጠገብ	6
	ቅፍርናሆም	
segment78.wav ከገሊላ ምድር ታላቅ	ምድር	7
segment79.wav የባህር መንገድ	የባህር	--
	መንገድ	68
segment80.wav ከሞተ ከተማ ለተቀመጡትም ዳሩ አላቸው ይጠመቁ ይፈጸም ዘንድ	ከሞተ/bemot	--
	ለተቀመጡትም	--
	ይፈጸም	--
	ዘንድ	62
segment81.wav ከይሁዳ ቃል አጠገብ ሲመላለስም ሁለት እንደ ማን	አጠገብ	6
	ሲመላለስም	--
	ሁለት	168
segment82.wav ቅዱስ የሚሉትን ልብስ ሆነ እንደ ሆነ መንፈስ ነውና አላቸው ገባ እሰጥሀለሁ	የሚሉትን	--
segment83.wav ታላቅ ማዶ ልብስ ነበር		
segment84.wav ደስ በኋላዬ ነው ሶስት ጋር ማዶ ተነስቶ ማዶ ጋር ቃል ወራት	በኋላዬ	2
segment85.wav ከዚያም ኢየሱስ	ከዚያም	12
segment86.wav የዘብዴዎስን ልጅ ያእቆብን ሁሉ	የዘብዴዎስን	---
	ልጅ	94
	ያእቆብን	2
segment87.wav ከእናቱ		
segment88.wav ድምጽ ምድር ታንኳይቱንና በታንኳ ከቶ ፈቀቅ በል	ታንኳይቱንና	----

segment89.wav ከእነርሱ		
segment90.wav የመንግስትንም ሁሉ ጊዜ ሰገዱለት	የመንግስትንም	---
segment91.wav ይህ ነው መልሶ ይህን ሁለት		
segment92.wav ከይሁዳ ሁሉ ደስታ ይልቅ ሰው በዳ ምድር ሁሉ	ሁሉ	188

Appendix F: STD Time with its Respective Transcription Wsing News Speech

segment04.wav**['0:00:30.596000', '0:00:36.730000']**segment04.wav ቀይ ፓዜጠኞች መካከል አብዛኛዎቹ ዛሬ ዛሬ ተገለጠ
segment05.wav**['0:00:37.621000', '0:00:39.513000']**segment05.wav ከዚህ ጋር የስራ ላይ ያሉት መሰረት ሁሉ ብቻ መሆኗም
segment06.wav**['0:00:40.583000', '0:00:48.523000']**segment06.wav ከዚህ ጦርነት ሁለት አለ
segment07.wav**['0:00:49.194000', '0:00:52.280000']**segment07.wav የኮሚሽኑ ዛሬ ና ነገር ግን ተቃውሞ
segment08.wav**['0:00:53.022000', '0:00:54.590000']**segment08.wav ችለናል በድመንና ሽራሮን በመውረር እውን ቃል
segment09.wav**['0:00:55.355000', '0:00:57.194000']**segment09.wav ታሪክ መቶ ና ላይ ናቸው ብዙዎችን ካሰ ዘመን
segment10.wav**['0:00:58.215000', '0:01:08.215000']**segment10.wav ባለፈው ሰኞ የቀሩት አዲስ ሌላ ሽልማት ሲሸለም በማለት
segment11.wav**['0:01:09.249000', '0:01:09.738000']**segment11.wav ተቃውሞ ልጅ ካሰ ስምንት ሮናልዶ ላይ ለማወቅ ተችሏል
segment12.wav**['0:01:10.408000', '0:01:11.648000']**segment12.wav ሌሎቹ በሙሉ ድምጽ ናቸው
segment13.wav**['0:01:12.403000', '0:01:14.160000']**segment13.wav ይህ ማህበር የስራ ዘመን በኋላ ግን መልኩ ነው
segment14.wav**['0:01:14.797000', '0:01:18.112000']**segment14.wav በኋላ ላይ ናቸው ያሉት አራት ሁለት ነገር ደግሞ በሰላም ገልጿል ሲል አለም ሙከራ ነበር ጽፈት ቤት መግለጫ አስታውቋል
segment15.wav**['0:01:19.026000', '0:01:24.086000']**segment15.wav ኳሶች ጋር የነበሩ በአዲስ ችለናል እጃቸውን ተችሏል የፖሊስ መግለጫው ነው ገልጿል
segment16.wav**['0:01:24.946000', '0:01:26.497000']**segment16.wav ተብሎ ቡድኑ ማስወገድ ነበረባቸው
segment17.wav**['0:01:27.122000', '0:01:33.914000']**segment17.wav ክብር ላይ አስታውቋል ቃል ለማቋቋም አለም ሰው አቶ ና ምጓጮች ከሰጡት ድጋፍ ጊዜው
segment18.wav**['0:01:35.022000', '0:01:44.899000']**segment18.wav ይህ ሁሉ የስራ ቃል
segment19.wav**['0:01:45.612000', '0:01:46.162000']**segment19.wav በዚህ ሌላ አጋጣሚ ግጭት ኤርትራ ተጠቃሚ ሚኒስትሩ በተለይ አስመራ ላይ ያሉት ግን ሰፊ እንደሆነ ገልጿል

Appendix G: STD Time with its Respective Transcription Using Bible Speech

segment00.wav**['0:00:04.573000', '0:00:08.812000']**segment00.wav ድረስ ሰው በይሁዳ ቤተ ልሄም በግብጽ በሄሮድስ ዘመን ደግሞ ልጇ ጊዜ
segment01.wav**['0:00:09.468000', '0:00:12.866000']**segment01.wav ጀምሮ ፈራ ሰገል
segment02.wav**['0:00:13.583000', '0:00:22.262000']**segment02.wav ሁሉ ድምጽ ራቅ አይተን ልገሰግድለት መጥተናልና ያለ
segment03.wav**['0:00:23.053000', '0:00:29.921000']**segment03.wav የሀዩኑን ሁሉ
segment04.wav**['0:00:30.596000', '0:00:36.730000']**segment04.wav ክርስቶስ ደግሞ ብሎ ልጅ ጠየቃቸው
segment05.wav**['0:00:37.621000', '0:00:39.513000']**segment05.wav ከይሁዳ ገዢዎች ከአባታቸው
segment06.wav**['0:00:40.583000', '0:00:48.523000']**segment06.wav ሀዝቤን ሰው ሁሉ የሚጠብቅ ራስህን ቃል ሰው ቃል ነው
segment07.wav**['0:00:49.194000', '0:00:52.280000']**segment07.wav ከዚህ ላይ ስለ ጋር ግን እርሱ ኮከቡ
segment08.wav**['0:00:53.022000', '0:00:54.590000']**segment08.wav ከጌታ ልጅ ደግሞ ዘመን
segment09.wav**['0:00:55.355000', '0:00:57.194000']**segment09.wav ከእነርሱ በጥንቃቄ ቀርበው
segment10.wav**['0:00:58.215000', '0:01:08.215000']**segment10.wav ከጌታ ልሄምም
segment11.wav**['0:01:09.249000', '0:01:09.738000']**segment11.wav ከእነርሱ
segment12.wav**['0:01:10.408000', '0:01:11.648000']**segment12.wav እርሱም ሰምተው ሄደ
segment13.wav**['0:01:12.403000', '0:01:14.160000']**segment13.wav ቃል ነው በምስራቅ ይልቅ ኮከብ
segment14.wav**['0:01:14.797000', '0:01:18.112000']**segment14.wav ራሄል ተቀመጥ ጽድቅን ተሰማ ከቶ ነበር
segment15.wav**['0:01:19.026000', '0:01:24.086000']**segment15.wav ሁሉ ከቶ ነው ቃል ጊዜ በታላቅ ደስታ እጅግ ደስ አላቸው
segment16.wav**['0:01:24.946000', '0:01:26.497000']**segment16.wav ከእናቱ ከማርያም ጋር ሂድ
segment17.wav**['0:01:27.122000', '0:01:33.914000']**segment17.wav ቃል ሰገዱለት
segment18.wav**['0:01:35.022000', '0:01:44.899000']**segment18.wav ኢየሱስ እንዳይመለሱ በል ከፍተው

Appendix H: Sample Code Display Integration with ASR

```
process.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {

            //START OF TRANSCRIPTION
            SpeechResult result;

            FileWriter fw = new FileWriter("C:\\Users\\user\\Desktop\\Integration\\transcription.txt");
            File folder = new File("C:\\Users\\user\\Desktop\\Integration\\autSegmented");

            File[] listOfFiles = folder.listFiles();

            for (File listOfFile : listOfFiles) {
                InputStream stream = new FileInputStream(listOfFile);
                StreamSpeechRecognizer recognizer1 = new StreamSpeechRecognizer(configuration);
                recognizer1.startRecognition(stream);
                if ((result = recognizer1.getResult()) != null) {
                    fw.write(listOfFile.getName() + "   " + result.getResult().getBestResultNoFiller() + "\n");
                }
                stream = null;
                recognizer1 = null;
            }

            fw.close();


    //END OF TRANSCRIPTION
```

Appendix I: Sample Code Display GUI.

```
//START OF GUI DEVELOPMENT
Font font = new Font("Nyala", Font.BOLD, 12);
JFrame frame = new JFrame("አማርኛ ንግግር ሰአት መፈለጊያ");
frame.setFont(font);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setPreferredSize(new Dimension(600, 700));


JButton brouse = new JButton();
brouse.setText("አማርኛ ንግግር ፋይል ምረጥ(ፏ)");
brouse.setFont(font);
brouse.setBounds(50, 50, 180, 30);

JButton segment = new JButton();
segment.setText("ፋይል ከፋፍል(ፃ)");
segment.setFont(font);
segment.setBounds(235, 50, 180, 30);

JButton process = new JButton();
process.setText("ፕሮሰስ ማድረግ ጀምር(ፆ)");
process.setFont(font);
process.setBounds(400, 50, 180, 30);
```

Appendix J: Browse Audio File Using the text-based STD System (GUI)