




Article

Amharic Speech Search Using Text Word Query Based on Automatic Sentence-like Segmentation

Getnet Mezgebu Brhanemeskel ¹, Solomon Teferra Abate ¹, Tewodros Alemu Ayall ^{2,3,*} and Abegaz Mohammed Seid ^{3,†}

¹ School of Information Science, Addis Ababa University, Addis Ababa 1176, Ethiopia

² Department of Computer Science, Zhejiang Normal University, Jinhua 321004, China

³ Department of Computer Science, Dilla University, Dilla 419, Ethiopia

* Correspondence: ayalltewodros@zjnu.edu.cn

† Current Address: Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha 34110, Qatar.

Abstract: More than 7000 languages are spoken in the world today. Amharic is one of the languages spoken in the East African country Ethiopia. A lot of speech data is being made every day in different languages as machines are getting better at processing and have improved storing capacity. However, searching for a particular word with its respective time frame inside a given audio file is a challenge. Since Amharic has its own distinguishing characteristics, such as glottal, palatal, and labialized consonants, it is not possible to directly use models that are developed for other languages. A popular approach in developing systems for searching particular information in speech involves using an automatic speech recognition (ASR) module that generates the text version of the speech where the word or phrase is searched based on text query. However, it is not possible to transcribe a long audio file without segmentation, which in turn affects the performance of the ASR module. In this paper, we are reporting our investigation on the effects of manual and automatic speech segmentation of Amharic audio files in a spiritual domain. We have used manual segmentation as a baseline for our investigation and found out that sentence-like automatic segmentation resulted in a word error rate (WER) closer to the WER achieved on the manually segmented test speech. Based on the experimental results, we propose Amharic speech search using text word query (ASSTWQ) based on automatic sentence-like segmentation. Since we have achieved lower WER using the previously developed speech corpus, which is in a broadcast news domain, together with the in-domain speech corpus, we recommend using both in- and out-domain speech corpora to develop the Amharic ASR module. The performance of the proposed ASR is a WER of 53% that needs further improvement. Combining two language models (LMs) developed using training text from the two different domains (spiritual and broadcast news) allowed a WER reduction from 53% to 46%. Therefore, we have developed two ASSTWQ systems using the two ASR modules with WERs of 53% and 46%.

Keywords: speech segmentation; spoken term detection; automatic speech recognition; manual speech segmentation



Citation: Brhanemeskel, G.M.; Abate, S.T.; Ayall, T.A.; Seid, A.M. Amharic Speech Search Using Text Word Query Based on Automatic Sentence-like Segmentation. *Appl. Sci.* **2022**, *12*, 11727. <https://doi.org/10.3390/app122211727>

Academic Editor: Kuei-Hu Chang

Received: 25 September 2022

Accepted: 14 November 2022

Published: 18 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The availability of a vast amount of information stored in audio and video repositories worldwide has increased the interest in searching on speech (SoS) [1]. SoS focuses on retrieving speech content from audio repositories that match user queries, i.e., searching for audio or speech by using any term of interest by text or segment of the audio or voice. Audio data includes everything that can be heard, such as speech, music, animal sounds, bell sounds, laughter, bird chirps, news footage archives, 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 50 Hz and 250 Hz,

whereas female pitch contribution is typically between 120 Hz and 500 Hz [2]. Audio files can be generated from different sources such as the internet and social media by individuals and organizations. Moreover, recent technological developments in the storage space of machines and their affordable prices make producing and storing audio files less challenging. Now, audio files can be recorded quickly and shared through many channels or platforms to reach the public immediately. These audio files are played using audio player software and social media such as YouTube. However, there is no way to search for a specific word in a speech on these audio players.

Speech segmentation is the process of decomposing a long speech signal into a shorter length. Speech segmentation can be classified as manual and automatic. Manual segmentation is a traditional approach in which trained phoneticians segment speech manually. However, this approach is uneven and time-consuming because it relies on hearing and visual interpretation of the necessary boundaries [2]. On the other hand, automatic segmentation segments speech automatically into sentence-like or phrase-like parts. This segmentation is convenient for the design of automatic speech recognition systems.

Searching on the speech of an audio file can be done by giving a 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 categories: keyword spotting, spoken term detection (STD), and query-by-example spoken term detection (QbE STD) [3]. Keyword spotting (word spotting) enables to locate a text query within a speech document or a speech stream [4]. In literature, the term keyword spotting has been used as STD. However, according to Larson et al. [4], in keyword spotting, the user query is known in indexing time, whereas in text-based STD the query term is specified at the search time. As a result, STD is more difficult to use since it has no prior knowledge of the queries that are being searched for [5]. Text-based STD is the process of locating a particular search term from a collection of segmented speech [6]. With the increased interest in SoS, STD is a type of SoS that helps to retrieve speech data by using text as a query word that represents a particular speech utterance [7]. The general structure and components of STD systems contain two steps to the STD procedure. The first stage, indexing, creates a database with an intermediate representation of the speech segments stored in a database. Using this intermediate database, the second stage is designed for locating putative occurrences of the query word. The search should be carried out quickly and precisely [5]. The STD has advantages including the possibility of retrieving any speech file that contains any term from its textual representation, allowing for an efficient search of any term in a large index. This technology can be accessed using any device with text input capabilities [3,7]. The state-of-the-art STD systems usually work 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 challenging since the LVCSR engine always misrecognizes the OOV words [8]. The OOV refers to words that are not in the lexicon and is the most common source of error in ASR [9,10]. Different approaches are available to minimize the OOV effect on the ASR. One way of achieving high lexical coverage is by building a language model (LM) on the morpheme level [11]. On the other hand, QbE STD is a technique in which a user presents the system with the desired audio snippets containing queries. The system then searches the database for segments that closely resemble the query [3].

ASR allows the machine to understand the user's speech and convert it into a series of words through a computer program; thereby creating a kind of natural communication between human and machine [12]. The ASR components include acoustic front-end, acoustic model, lexicon, LM, and decoder. The acoustic front-end converts the 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 called feature extraction. Feature vectors are generally generated every 10 milliseconds using a 25 millisecond overlapping analysis frame. The decoder searches through all possible word sequences to find the sequence of words most likely to generate. The LM is generally an n -gram model where each n word's

likelihood is solely dependent on its $n - 1$ predecessors [13]. Moreover, smoothing is a technique essential in constructing the n-gram LM, a main part of speech recognition. It is a set of procedures for fine-tuning the maximum likelihood estimation (MLE) by counting events in the training corpus to produce more accurate probabilities. Some of the smoothing techniques, such as Laplace Smoothing, Add λ Smoothing, Natural Discounting, Good-Turing Smoothing, Interpolation, and Backoff can solve the problem of data sparsity based on the raw frequency of n-grams. The details of each smoothing technique are clearly elaborated by Tachbelie [14]. The author used interpolation as a smoothing technique and solved the problem of data sparsity through n-gram hierarchy. Furthermore, the probability estimates of all n-gram orders were combined based on the assumption that, if there is not enough data to estimate a probability in the higher-order n-gram, the lower-order n-gram can frequently give relevant information [14].

To date, around 7000 languages are spoken in the world [15]. Amharic is the official working language of the government of Ethiopia, an East African country with a population of over 100 million. It is one of the Ethio-Semitic languages, which belongs to the Semitic branch of the Afro-Asiatic family and has the second largest number of speakers in the world after Arabic [16] and is the most widely spoken Semitic language in Ethiopia [17]. The majority of the speakers of Amharic can be found in Ethiopia; however, there are some speakers in other nations, such as Israel, Eritrea, Canada, the USA, and Sweden [9,18]. Audio data are abundantly found in Amharic language via the various private, social, and government media platforms. The high prevalence of social media and multimedia in our interconnected global society today has created the need to access different audio files. Individuals and organizations use this audio file to satisfy their information needs. Locating a spoken word in an audio file, however, is a challenge, because users may know that the speaker spoke a word but may not know in which part of the audio file that the word was spoken. For instance, to find 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 location of spoken word ኢትዮጵያ within that audio file. Therefore, automatically locating 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 properties. 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 [19]. Consequently, it is not possible to directly use models implemented for other languages. Currently, the different tools are available to search text; however, they are not applicable for speech or audio search [20]. Although some online web applications allow users to convert a given audio file to an English text, the resources, technology, and research on speech searching remain in their infancy [6]. In addition, Chaudhary et al. [21] studied keyword-based indexing of a multimedia file in the English language to allow users to search for a particular spoken speech using text and display the time frame and the utterance. However, research has not been done for searching or locating the spoken word (utterance) time frame interval from the audio file in any Ethiopian language. Therefore, the result of this research will enhance and add additional features to audio information retrievals.

The contributions of this research work include:

- Since there is no prior research made on the effect of automatically and manual segmented speech on Amharic ASR, we compared the performance of ASR using automatically and manually segmented test speeches;
- We propose Amharic speech search using text word query (ASSTWQ) based on automatically sentence-like segmentation and using a previously developed speech corpus, which is in a broadcast LVCSR domain, together with the in-domain using the Bible domain speech corpus;
- We showed the effect of ASR recognition errors on searching the recognized text and the effect of ASR recognition errors on searching using different domains;

- We prepared the speech corpus and conducted an extensive experimental evaluation to check the performance of the proposed work.

The rest of the paper is described as follows: Section 2 presents a discussion of the Amharic language and Section 3 elaborates the related works. Section 4 describes the materials and methods. The experiments are provided in Section 5. The results and discussion are provided in Section 6. Finally, the conclusions are summarized in Section 7.

2. Characteristics of Amharic Language

This section presents the characteristics of the Amharic language. This language has its own writing system and distinguishing characteristics from other languages by phonology, consonants, vowels, and Amharic morphology [17].

2.1. Amharic Writing System

Unlike other Semitic languages such as Arabic and Hebrew, Amharic is written from left to right. Present-day Amharic has acquired its composing framework from Ge'ez /gə'əzə, which is still the classical and ministerial dialect of Ethiopia and uses a grapheme-based writing system called Fidel /fidalə/ [9,18]. Amharic symbols are categorized into four categories consisting of 276 distinct symbols; these are core characters, labiovelar, labialized, and labiodental. A sample list of Amharic core characters is shown in the Appendix B.

2.2. Amharic Phonology

The study of speech sounds used in various worldwide languages is known as phonetics [19]. Amharic has 31 consonants, which are generally classified as stops, fricatives, nasals, liquids, and semi-vowels. The sounds that are not found in English, such as ጸ[px], are glottalized sounds. The existence of palatal consonants such as ሸ[sx] and dental consonants such as ቸ[t], labialized consonants that are pronounced by a slight round of the lips ገ[w], loan words ሸ[v] and the existence of geminated words make the language distinctive from any other language. The Amharic language has 38 phonemes, including 7 vowels and 31 consonants.

2.3. Consonants

Out of the 31 Amharic consonants, a few of the Amharic consonants have similar phonetic transcriptions to English. These include ብ[b], ድ[d], ፍ[f], ግ[g], ሀ[h], ክ[k], ለ[l], ሞ[m], ን[n], ጥ[p], ር[r], ሰ[s], ቸ[t], ሸ[v], ወ[w], ደ[y] and ገ[z]. They correspond to the English consonants b, d, f, g, h, k, l, m, n, p, r, s, t, v, w, y, and z, respectively. In addition, there are consonants that sound the same as English sounds but are represented using different symbols. These symbols includes ጸ[ch], ጻ [nx], ሸ[sx] and ጸ[zx]. Sounds that are characteristic of Amharic but that are not found in English are ጸ[px], ጥ[tx], ፅ[xx], ፍ[cx], and ቸ[q] [17,22]. Categories of Amharic consonants are indicated in Appendix A.

2.4. Vowels

The Amharic language 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 1 [19] 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) [19].

Table 1. Categories of Amharic vowels.

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

2.5. Amharic Morphology

Morphology studies word forms in terms of morphemes, which are the smallest semantic grammatical units [23]. 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 vowels inserted among the root consonants [19,24]. The Amharic language words 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 the speech or lexical category from one to another. For the word teach by adding er, we can get teacher, in which, 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 [19,24]. By adding a suffix to the stem ሰብር it forms words such as ሰብር-ኩ [I broke], ሰብር-ን [we broke], ሰብር-ሽ (feminine second person)[you broke], and the immediate object is identified as ሰብረ-ኝ [he broke me] as is pointed out by Abate [19].

3. Related Work

In this section, previous studies related to searching for specific utterance using written text as query word and speech segmentation on ASR are explored.

3.1. Speech Segmentation

Meinedo et al. [25] developed an ASR system for automatic speech transcription of broadcast news (BN) in Portuguese language. To develop the system, the authors followed a hybrid approach combining hidden Markov model (HMM) and multi-layer perceptron (MLP). The system was tested with a 29-minute test set speech and compared with 3 transcription results. The first test was done on 241 manually transcribed sentences, and the second by considering the whole program as one sentence where no preprocessing was made, while the last test set was segmented automatically to produce 366 sentences. The recorded word error rate (WER) for the first, second, and last test sets were 26.9, 27.1, and 29.0, respectively.

Tamiru and Abate [22] has developed a sentence-level automatic speech segmentation system for Amharic, which is used to segment the spoken speech into sentences. To implement the sentence-like segmentation, the author used two approaches. In the first approach, an automatic tool for segmenting and labeling Amharic speech data were used. In this approach, preprocessing rule-based segmentation using Audacity software, was applied to segment an audio file and given to the different acoustic models. Then, the audio file with its respective transcribed file was given to the Forced aligner, and the segmentation result was displayed. In the second approach, the author used energy and F0 features 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 were used to check the performance and accuracy of the supervised classifier. Following the two different approaches, the author found that the results of the first approach (rule-based) were better than the second approach.

Kim et al. [26] compared the relative performance of ASR systems developed using automatically and manually transcribed speech corpora. They used two sets of manual transcriptions and five sets of automatic transcriptions (Google Cloud, IBM Watson, Microsoft Azure, Trint, and YouTube). The obtained results showed that manual transcription is better than all other transcriptions (Google Cloud, IBM Watson, Microsoft Azure, Trint, and YouTube), even though YouTube offers accurate transcription compared to others.

Heerden et al. [10] made an experiment on sub-word unit syllable-like and morpheme-like units for different languages. The aim of their research was on how to reduce OOV keywords that are generated by the ASR. The researchers compared 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), and Pashto (Eastern Iranian) languages. The corpora used in their experiments were the “Full language packs”, which were distributed in the fourth year of the intelligence advanced research projects activity (IARPA) project BABEL and each contained about 40 h of training data for all languages. Kaldi was used as a speech recognition toolkit. From the results they obtain when comparing the OOV results, whether to use syllables or morphemes becomes a language-specific choice. For Amharic, Dholuo, and Pashto, the morpheme-based OOV results were found to be best, while for the rest of the languages, syllable-based OOV results were slightly better (Guarani, Igbo, Javanese) to significantly better.

Sharma and Rajpoot [27] segmented the speech signal into silence, voiced, and unvoiced regions. 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) and short time energy (STE), which helps to quantify how much energy is in a speech signal at any given moment: high for voiced, low for unvoiced, 0 for silent, and Fundamental Frequency (F0). The algorithm was applied to 15 selected Hindi words spoken by 4 persons (3 male and 1 female) and each was spoken 3 times. By using 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 silent regions in the word, and then dividing that number by the total.

Silber-Varod et al. [28] 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, using existing tools, and to explore their potential to provide good enough—however not perfect—video transcriptions. In solving their stated hypothesis, they used the already available speech recognition models, Google/HTML5 speech recognition system for Hebrew and nuance mobile developer program (NDEV). A total of 40 min of Hebrew speech was used for their ASR experiment by using the above ASR engines. From their first experiment, they found that the word recognition rate (WRR) tests showed that the ASR’s performed better with read speech than with lectures, in both quantity and quality.

3.2. Searching on Speech

Chaudhary et al. [21] presented an interactive media player that enables the user to perform offline audio content-based searching capabilities with a given multimedia file. The research objective was to alleviate the problem of massive online courses (MOOCs), a free online course available for anyone to enroll, but in which only 15% of users complete the course.

Hassen and Atnafu [29] proposed an Amharic speech search engine by designing an Amharic speech document. Since most search engines are designed for English, they struggle to find documents written in Amharic. The prototype developed by the author has four essential components: crawler, audio processing, indexer, and query engine. The experimental results revealed that the Amharic speech retrieval engine had an accuracy of 80% on the top 10 results and a recall of 92% as compared to its corresponding retrieval engine.

Laham et al. [30] proposed a system that can automatically process a YouTube video file. It allows users to identify the discussed topics in a significantly shorter time through a search engine. The topic of interest is described by keywords used for searching the video or audio material. The proposed system works in a way that first checks the coming multimedia data types and their file extension. If it is a video file in *.wav format and it is automatically transcribed to its respective English text. From the transcribed text, the respective topic can be deduced. After transcription, the indexed topic with the predefined list of topics (keyword spotting) is matched. Then, the matching results from the previously mentioned process and the respective video file are stored in the database, which

can be used by the users while requesting the system by entering text query word. The transcription accuracy of the proposed system was 50–80%.

The other research made by Arisoy [31] was the Turkish Broadcast News transcription and retrieval system. Developing the OOV was a challenge where even sub-word-based recognition units were utilized. To alleviate this problem and increase the accuracy, the researchers used moderately sized vocabularies, which performed better for a vocabulary size of 500 k. The researchers developed a Spoken Term Detection system and a Spoken Document Retrieval system. To retrieve the spoken data, ASR was used.

In the present study, we aim to achieve two purposes: First, to check the performance of an ASR system that is developed using automatically segmented audio files on automatically segmented test speeches. Second, to investigate the performance of searching a text word and displaying the time interval in which the word is spoken in a speech file.

4. Materials and Methods

4.1. Materials

This section presents the speech corpus we used for comparing the performance of ASR/LVCSR on manually and automatically segmented speech and the proposed ASSTWQ. The training and test sets (corpus) for performance comparison were prepared from the spiritual domain. For the test speech, we used similar speeches, but we segmented them manually and automatically. On the other hand, the speech corpus we used for the development of proposed ASSTWQ was from spiritual and broadcast domains.

4.1.1. Training Speech Corpus

We developed the baseline ASR system using a 1 h and 35 min in-domain training speech for the acoustic model and 1050 text sentences, which is the transcription of the training speech, for the LM. This corpus was downloaded from a publicly available YouTube audio. These sentences were obtained after automatically segmenting the audio using a value of 600 for minimum silence and a silence threshold of -35 and transcribed accordingly.

4.1.2. Test Speech Corpus

We prepared two different test speech corpora: manually 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 30-min spiritual speech was used for manual and automatic speech segmentation. We prepared two kinds of automatically segmented test sets: sentence-like segmentation and word/phrase like segmentation.

Manual Segmentation (Mseg)

The manually segmented speeches were taken from the study done by Tamiru and Abate [22]. Even if we found a lot of manually segmented speech from the indicated source, we used 26 min and 30 s of the segmented speech and we manually segmented and added an additional 3 min and 30 s unsegmented speech, to a total of 307 segments. Since the sample rate of the manually segmented speech was 22,050 Hz, we converted it to 16,000 Hz due to our CMU Sphinx configuration. We also further manually segmented the text to be aligned with the segmented speech.

Automatic Sentence-like Segmentation (A-I)

The automatic segmentation of unsegmented speech is done using silence. A 30-min unsegmented speech was segmented automatically using parameters of minimum silence of 600 and a silence threshold of -35 . We segmented the audio files according to the minimum silence and threshold parameters into 369 segments. Once we segmented automatically, the next step was to transcribe the automatically segmented speech. We annotated the respective text from an existing unsegmented long text that was available in WordProject [32] by listening to the start and end of the automatically segmented speech.

Automatic Word/Phrase-like Segmentation (A-II)

The automatic segmentation of unsegmented speech was done using silence. A 30 min unsegmented speech is segmented automatically using parameters of a minimum silence of 400 and a silence threshold of -26 . A total of 877 segmented *.wav files were obtained by using the minimum silence and threshold. These segmentation were mostly phrase and word-level segmentation. The transcription procedure that was used for sentence-like segmentation was repeated for the phrase-like segmentation.

4.1.3. Dataset to Train the LM

The other component of ASR is the LM. We collected a total of 10,602 training sentence for LM (TSLM) from the Holy Bible [33] and WordProject [32]. We also used well-constructed sentences from Ref. [22] for our LVCSR development. Here, we tried to read available online resources and segment them in a way that will give meaning when they are read as a sentence. All unnecessary characters and punctuations are removed. We have also experimented with the use of a closed and open vocabulary LM.

4.1.4. 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) [34]. For example, the phoneme for the word ሀገር (country) are ሀ (h a) ገ (g aa) ር (r ee). Using the above rules for every characters we convert the word to its respective phone. Therefore, the phone for the word ሀገር will be (h a g aa r ee). In the present study, GTP conversion was made based on the phonetic dictionary by Abate et al. [35]. A total of 6446 words with their respective phonemes were used for the ASR development using the spiritual speech.

4.1.5. Graphemes Normalization

The other preprocessing task we did on the training and test transcription was replacing characters whose sounds were identical but had different shapes [36]. So, in our case, we used graphemes (ሀሁሂሃሄህሆ) instead of using (ሐሑሒሓሔሕሖ) and (ሰሱሰሰሰሰሰሰ) instead of using (ሠሡሢሣሤሥሦሧ) because both have the same sound.

4.1.6. Corpus Used for ASSTWQ Development

The speech corpus we used to develop ASSTWQ's acoustic model was the news read speech corpus developed by Abate et al. [35], and the spiritual speech corpus developed by Tamiru and Abate [22]. From the first corpus of 100 speakers, we used a total of 72 speakers and trained our acoustic model. The test speech was also directly used from the same source. To this end, we have used very large speech (for acoustic modeling) and text (for language modeling) corpora developed by Abate et al. [35]. As a result, we developed two acoustic models, first using the LVCSR broadcast domain and evaluated using spiritual (Bible test set) and broadcast (news test set) domains. The other acoustic model was developed by combining the training set of the two corpora (spiritual and broadcast) domains and evaluated using the test sets/speeches of the spiritual domain (Bible test sets). The length of the unsegmented audio file we used for the evaluation was 8 min and 31 s for the spiritual domain. This speech was read by a woman in a way to simulate a news reading. The texts were taken from the read speech corpus of Abate et al. [35]. On the other hand, we used 8 min and 31 s publicly available YouTube read speech for the system that was developed using the spiritual domain. The LM used in the development was achieved by combining a LM with 10,602 sentences from the spiritual domain and broadcast domains used by Abate et al. [35].

4.2. Methods

We followed four steps to achieve our research objective. The first step was the development of the ASR using in-domain training speech corpus for the acoustic modeling. Then, the second step was to check the performance of the developed ASR for selecting

the optimal automatic speech segmentation. In the context of our work, optimal segmentation is defined as the lowest WER obtained compared to the WER obtained to the manually segmented test speech. The third step was to develop another acoustic model using LVCSR [35]. In the final step, we used the acoustic models of the ASR for text-based STD development.

4.2.1. ASR/LVCSR Development

According to the classical structure of ASR development, three models were developed as components of the ASR that do the speech transcription. These are acoustic, language, and lexical models [37]. The decoder of the ASR system uses these three developed models to search through all possible word sequences and find the sequence of words that is the most likely the transcription of the input speech. This process of establishing statistical representation is done on the feature vector sequences that are computed from the input speech waveform. The block diagram in Figure 1 depicts the developed ASR model that was trained and checked the performance using manually and automatically segmented test speech corpus.

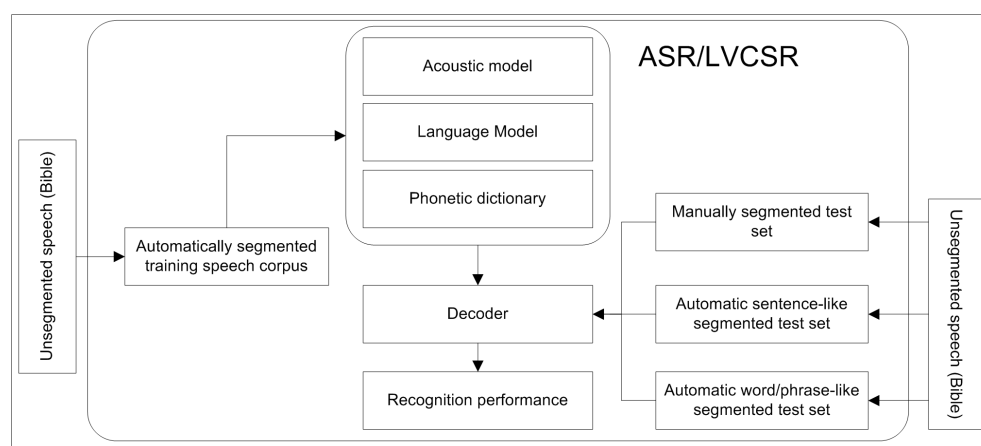


Figure 1. ASR development using automatically and manually segmented speech corpus.

4.2.2. Proposed ASSTWQ

We propose Amharic speech search using text word query (ASSTWQ) and based on automatic sentence-like segmentation. We used previously developed speech corpus, which is in a broadcast LVCSR domain, together with the in-domain using the Bible domain speech corpus. In order to develop ASSTWQ, we followed the general architecture in Ref. [5]. One of the essential components of the text-based STD is the ASR that produces the transcription of the speech. The main difference between text-based STD and other speech search 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. The proposed ASSTWQ architecture is shown in Figure 2, which depicts how the speech search system was implemented. First, we developed HMM-GMM acoustic models using in- and out-domain training speech corpus in different ways. We used two better performing acoustic models for the development of the text-based STD system. On the text-based STD, speech search begun from the unsegmented speech in the form of a *.wav file. Then, the unsegmented speech was automatically segmented into sentence-like segmentation. This was segmented based on the threshold and minimum silence specified within the system. Then, this automatically segmented speech was decoded into text/decoding. In order to make the alignment process easier, this segmented speech file name was extracted and maintained. The other task was finding the time frame upon which the segmented speech was located. Finally, file concatenation/alignment/indexing was made. Here, we first combined the segmented file with the respective time frame and then, using this result, we concatenated it with the decoded text. All the generated files were maintained in a Windows file system

that 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 [35]. The unsegmented speech was converted into *.wav since the ASR components of text-based STD could transcribe formats with *.wav.

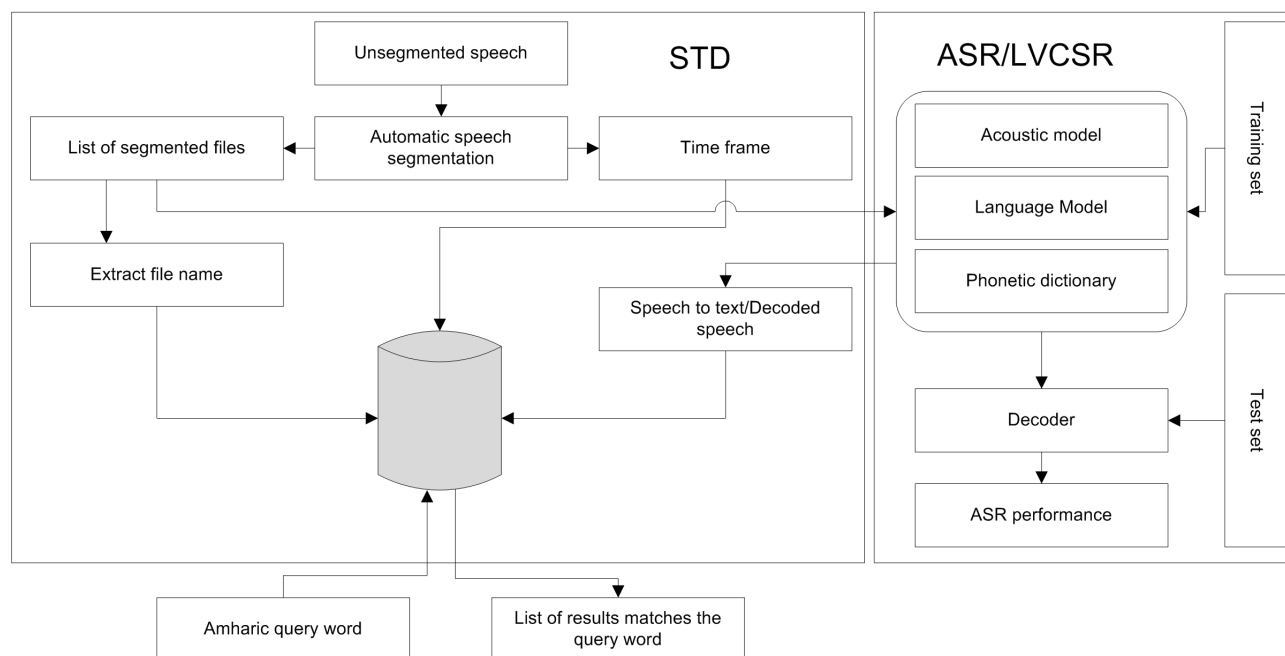


Figure 2. The proposed ASSTWQ architecture.

5. Experiments

5.1. Experimental Setup

We used the following experimental setup to test the proposed work. 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 (LENOVO, Hong Kong, China). This was developed using the popular CMU Sphinx toolkit that includes various libraries for creating speech applications [37]. To train our acoustic model, we were required to prepare a file dictionary, phone, filler, LM, transcription, and the list of the wave files. The acoustic model was developed using the triphones HMM structure. Since we trained continuous models for a large vocabulary, we used a Gaussian number of 8. The acoustic front-end was made using mel-cepstrum MFCC features with noise tracking and spectral subtraction for noise reduction used by the recognizer. The acoustic model that was developed for the comparison of automatically and manually segmented test speech (Bible speech corpus) was done using a grapheme pronunciation dictionary, which was created by writing a simple python script. The LM was developed using the SRI language modeling toolkit (SRILM) as a tool. Using this tool, 3-gram language modeling was developed. Manually segmented test speeches were prepared using the audio processing software Audacity [22] whereas automatic segmentations were made using python's pydub package. This package is a speech segmentation package based on silence detection on the energy [38].

On the other hand, the text-based STD was done on Windows 10 pro. We used Netbeans 8.0.2 for java development and also implemented python in an anaconda environment. One of the components of the STD system was used to segment the given unsegmented audio file into a list of automatically segmented audio files using the pydub package. The other component that was used in text-based STD was the LVCSR's acoustic model, which was developed using the CMU Sphinx toolkit. Two acoustic models were developed; the first using the broadcast domain and the second by combining the two speech corpora (the news and the Bible).

The LM for both acoustic models were developed using SRILM, but the LM that is used with the second acoustic model was developed by combining two LMs (the news and the Bible) using different lambda values [0.1-1]. The time frame of each segment is located using the python's pydub library. By using the acoustic model, LM, phonetic dictionary, and Sphinx4 API, automatically segmented speech was decoded. The alignment of the time frame and decoded text was made using java and python in combination and stored in the Windows file system. Therefore, the file was easily searched using a simple Java string mapping function that compares query terms and the aligned text and retrieves the time with the transcribed text. The search results were displayed using a graphical user interface (GUI) that used Java's JFrame, which is a top-level container that provides a window on the screen.

5.2. Evaluation Metrics

We measured the performance of ASR and STD. The accuracy of the ASR can be measured using word error rate (WER). The WER is the most widely used metric [12,39] and is defined in Equation (1) as:

$$WER = \frac{Insertion(I) + Substitution(S) + Deletion(D)}{No. of ReferenceWords(N)} * 100, \quad (1)$$

where S refers to 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 lower value of WER indicates a better ASR model.

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). The ATWV is a new metric created to reflect one potential use of an STD system. It is used to quantify the system accuracy on a particular set of query words [40]. ATWV is defined in Equation (2) as:

$$ATWV = mean \left(\frac{N_{correct(s)}}{N_{true(s)}} - \beta \cdot \frac{N_{spurious(s)}}{T - N_{true(s)}} \right), \quad (2)$$

where the search term (s) occurs $N_{true(s)}$ times in the reference transcript and the system makes $N_{correct(s)}$ correct and $N_{spurious(s)}$ 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 $N_{true(s)}$ is positive. The higher the value of ATWV indicates the better STD.

6. Results and Discussion

6.1. Sample Segmented Test Speech

The 30-minute unsegmented Bible speech was segmented manually and automatically to check its effect on the performance of ASR. Table 2 shows the comparison between manual segmentation and the two automatic segmentation's: A-I and A-II, in terms of the duration of each segment. From Table 2, it can be seen that A-I is closer to human segmentation Mseg while the duration of A-II is smaller than both the Mseg and A-I.

Table 2. Sample segmented files and transcriptions.

Segmented Audio	Mseg		A-I		A-II	
	Annotated Text	Duration	Annotated Text	Duration	Annotated Text	Duration
1	ኢየሱስም በይሁዳ ቤተልሄም በንጉሱ በሄሮድስ ዘመን በተወለደ ጊዜ እነሆ ሰብአ ሰገል የተወለደው የአይሁድ ንጉስ ወዴት ነው	00:08	ኢየሱስም በይሁዳ ቤተልሄም በንጉሱ በሄሮድስ ዘመን በተወለደ ጊዜ	00:04	ኢየሱስም በይሁዳ ቤተልሄም በንጉሱ በሄሮድስ ዘመን በተወለደ ጊዜ	00:04
2	ኮከቡን በምስራቅ አይተን ልንሰግድለት መጥተናልና እያሉ ከምስራቅ ወደ ኢየሩሳሌም መጡ	00:06	እነሆ ሰብአ ሰገል የተወለደው የአይሁድ ንጉስ ወዴት ነው	00:03	እነሆ ሰብአ ሰገል	00:01
3	ንጉሱ ሄሮድስም ሰምቶ ደነገጠ	00:02	ኮከቡን በምስራቅ አይተን ልንሰግድለት መጥተናልና እያሉ ከምስራቅ ወደ ኢየሩሳሌም መጡ ንጉሱ ሄሮድስም ሰምቶ ደነገጠ	00:08	የተወለደው የአይሁድ ንጉሥ ወዴት ነው	00:01
4	ኢየሩሳሌምም ሁሉ ከእርሱ ጋር	00:02	ኢየሩሳሌምም ሁሉ ከእርሱ ጋር የካህናትንም አለቆች የሕዝቡንም ጻፎች ሁሉ ሰብስቦ	00:07	ኮከቡን በምስራቅ አይተን ልንሰግድለት መጥተናልና እያሉ	00:03
5	የካህናትንም አለቆች የህዝቡንም ጻፎች ሁሉ ሰብስቦ ክርስቶስ ወዴት እንዲወለድ ጠየቃቸው እንዲወለድ ጠየቃቸው	00:07	ክርስቶስ ወዴት እንዲወለድ ጠየቃቸው እነርሱም አንቺ ቤተ ልሄም የይሁዳ ምድር	00:06	ከምስራቅ ወደ ኢየሩሳሌም መጡ	00:01

6.2. Comparison of Segmented Speech Using Varying LM

Performance comparison of manually and automatically segmented test speeches by using different LMs having a varied size and using the open and closed vocabulary technique was made. Figure 3a,b show mainly the effect of the changing LM using open and closed vocabularies on the manually and automatically segmented speeches. While increasing the size of training text for the development of the LM from 5 k to 10.6 k sentences, the performance of ASR is also increased. When they are compared in terms of their WER, the systems achieved closer results on the Mseg and A-I test sets. However, the performance of the systems on Mseg and A-II was different. In addition, using closed vocabulary in the LM showed a major decrease in the WER when it is compared to using an open vocabulary.

The language models' quality was evaluated first in terms of their perplexity [14]. The perplexity for LMs obtained using 5000, 10,000 and 10,602 sentences were 6233.63, 4217.63, and 1486.44, respectively. In addition to the perplexity of the LM's, we also computed the OOV rate and vocabulary size for the training text of 5000, 10,000, and 10,600 sentences. Table 3 shows the OOV rates of the LM that was trained with different training (5000, 10,000, 10,600) sentences. The result shows that as the size of the vocabulary increases, the OOV

rate decreases and the better the LM is. Therefore, the size of the vocabulary increases the performance of ASR. This is because ASR/LVCSR works with a phonetic dictionary. If these words are not in the lexicon, then the word is considered as out-of-vocabulary (OOV) which is one of the main sources of error in automatic speech recognition. Therefore, as the OOV rate increases WER also increases.

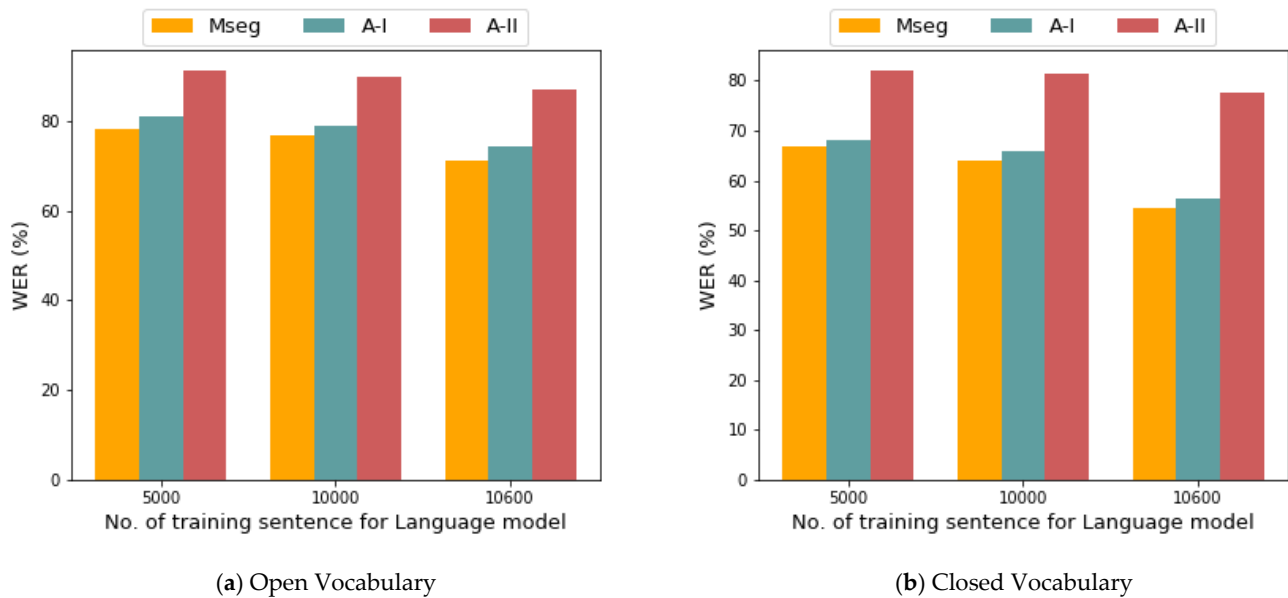


Figure 3. The effect of the LM on the ASR.

Table 3. OOV rate and unique vocabularies of the LM Mseg, A-I, and A-II as the test text.

No	LM	No.TSLM	Unique Vocabularies	Test Set			OOV Rate		
				Mseg	A-I	A-II	Mseg	A-I	A-II
1	OOV5000LM	5000	13,573	90	91	245	218	219	222
2	OOV10000LM	10,000	25,673	90	91	245	153	154	158
3	OOV10600LM	10,600	26,364	90	91	245	100	101	103

No. of training sentence for Language model (No. TSLM).

6.3. Comparison of Segmented Speech Using Training Set

The performance of the ASR system was compared by varying the training sets and keeping the test transcriptions the same for every iteration. This was conducted so to check how the ASR performs while changing the training set and its effect on the automatically segmented speeches. Figure 4 shows how the performance of ASR varies while changing the training set, keeping the LM and test speech the same, and using similar vocabulary. The total sentences used to create a LM were 10,602. All characters used in the test transcription were also available in the training set. From the result in Figure 4, the bar chart of A-I and Mseg shows a very slight difference in WER. As the WER of the Mseg decreases, the WER result of the A-I also decreases, and vice versa. However, the WER result of A-II was very large when compared to that of the Mseg in every experimental iteration. This implies that sentence-like automatic segmentation of A-I would yield a very close result to Mseg. On the other hand, A-II shows a very large difference with the Mseg in every experimental iteration.

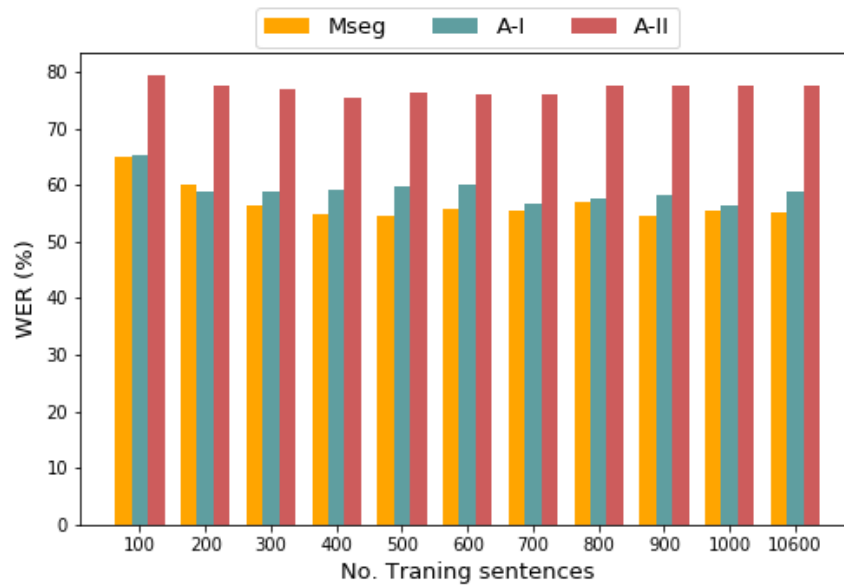


Figure 4. WER of Mseg, A-I, and A-II by varying the training sets

6.4. Comparison of Segmented Speech Using Test Sets

The comparison was made to check the performance of ASR by varying the length of the test transcription using acoustic models developed on the training corpus in every experimental iteration. In the experiment even if the length of the sentence is different the content of the speech (starting and ending) is the same for the test transcription. Figure 5a–c show the result of ASR performance using the open vocabulary LM, before and after graphemes normalization of the training and test sets. As a result of grapheme representation in the pronunciation dictionary, by using the same pronunciation for all words that sound the same, such as *ah*, (h a) and *ʔ* (h a) with *u* (h a), we gained a WER reduction of 2.5% for Mseg at segment 87, 2.4% for A-I at segment 91, and 1.24% for A-II at segment 245, and other segments also show a decrease in WER. The results show that both the manually and automatically segmented test speeches have a slightly lower WER. In addition, in all experiments, the speech segmentation effect of A-I shows a closer WER difference to that of Mseg. However, when we comparing A-II with Mseg, A-II showed a relatively high WER difference with and without graphemes normalization.

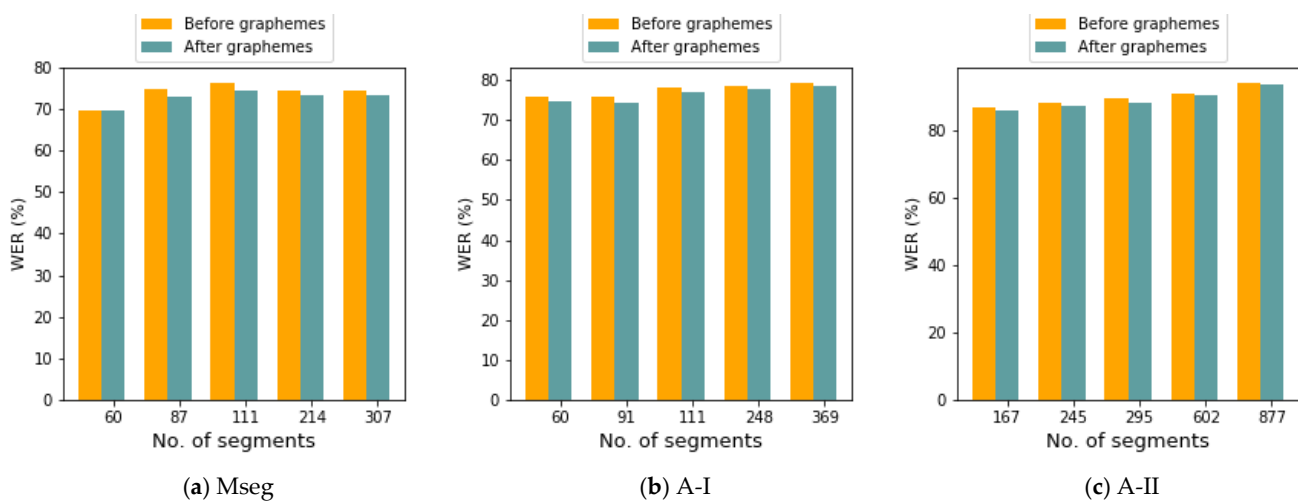


Figure 5. WER of Mseg, A-I, and A-II before and after graphemes normalization.

6.5. Segmentation Time

While performing preprocessing, we also tried to check for the time taken to segment an unsegmented audio file. It took an average time of 3 h to segment a 3 min and 30 s speech into 40 chunks or segments even if the length of the unsegmented file is short. The tasks done in Audacity included naming the file according to the user's interest, which may result in error and overcomplication. However, performing automatic segmentation, for a 30 min and 30 s file, took us 3 min. This was a favorable result when compared with the manual segmentation. Therefore, the result can be used as a baseline to forecast when the audio file is large.

6.6. The Effect of Different Domains on LVCSR

Table 4 shows the performance of the LVCSR models developed using out-domain and in-domain training speech evaluated on different test speech domains. The results show that the models developed using out-domain training speech corpus gained lower WER on the test speech of the same domain, compared to their performance on the Bible test speech corpus. Even though using a closed vocabulary decreases the WER, we did not observe a significant difference between the two domains. Therefore, from the result, we selected models that gain the minimum WER for our system development. We developed ASSTWQ with LVCSR's WER of 53.6, which has the best performance for the news domain.

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

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

6.7. Performance of LVCSR Using LMs of Different Domain

This section shows the performance of the LVCSR obtained by using LMs developed by combining LMs trained on the training text corpus of different domains (the Bible and news corpus). The Bible speech corpus was used to test the model. This experiment was conducted because the WER of LVCSR was 90.6%, as depicted in Table 4, using Bible text corpus on LVCSR trained using news speech corpus. We therefore conducted another experiment to check the performance of LVCSR by combining the LMs that are trained on the Bible training text corpus and the Bible text. The result was obtained by changing the lambda values with an interval of 0.1 and using the same training and test speech. We obtained a result of 46% for all experiments except for one, in which we found 47%, where the lambda value was 1. From the two experiments (see the results shown in Tables 4 and 5), 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, the WER of 90.6% was obtained by using LVCSR news speech and the Bible as the test speech. By using the same Bible test speech, but by combining the two different domains of news and Bible speech and interpolating the LMs of LVCSR news and Bible LMs, we found a WER of 46%. Therefore, combining two different training speech corpus and LMs would yield a decrease in WER. Therefore, we have used this ASR with a WER of 46% to develop the Amharic text-based STD, which is applied on the Bible domain. In addition, we combined the training speech corpus of the LVCSR and Bible speech corpus and interpolated the LMs of the LVCSR and the Bible speech, and we experimented with it using the Bible speech corpus, which resulted in a slight reduction in the WER.

Table 5. Using an interpolation technique to reduce the WER of the speech recognizer.

Training Text	Test Speech	λ (Lambda) Bible	WER (%)
Bible+News speech corpus	Bible speech	0.1	46
Bible+News speech corpus	Bible speech	0.2	46
Bible+News speech corpus	Bible speech	0.3	46
Bible+News speech corpus	Bible speech	0.4	46
Bible+News speech corpus	Bible speech	0.5	46
Bible+News speech corpus	Bible speech	0.6	46
Bible+News speech corpus	Bible speech	0.7	46
Bible+News speech corpus	Bible speech	0.8	46
Bible+News speech corpus	Bible speech	0.9	46
Bible+News speech corpus	Bible speech	1.0	47

6.8. Evaluation of the Proposed ASSTWQ

6.8.1. Speech Search Using News Data

The screenshot shown in Figure 6 depicts Amharic text-based STD GUI, which can accept Amharic query text using an unsegmented Amharic audio file from the broadcast domain and display where the speech is located in hours, minutes and seconds. The starting and ending times are displayed within the square bracket, as shown in the text area of text-based STD GUI. The following steps are to be followed to search for 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, the user can select a button labeled አማርኛ ንግግር ምረጥ(፩) (select Amharic speech file) that shows list of directories for the file to locate the time frame. After the user selects a file, the user clicks a button labeled ፋይል ከፋጭል(፪) (segment file). In this step, the system will internally segment the audio file that is already selected in the first step. Segmentation of the audio file is made automatically according to the parameter given or by using a minimum silence and silence threshold.

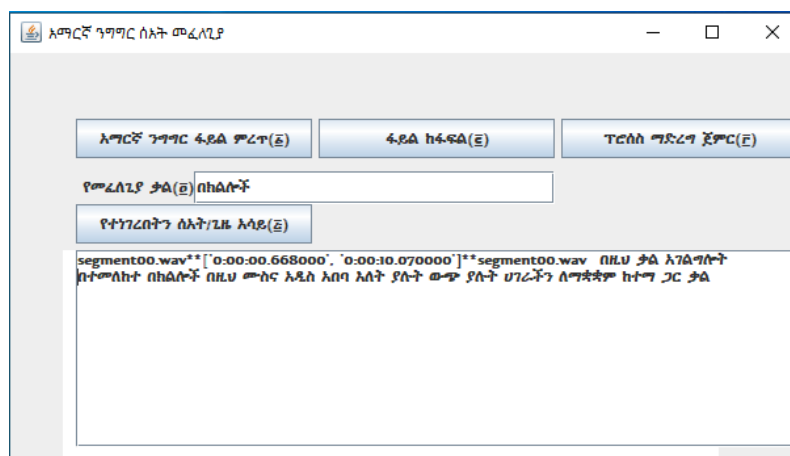


Figure 6. GUI for searching the speech using different news query word.

After segmentation is completed, the third step is the basic process where the time frame of every segmented speech is located or identified, and where the transcription and alignment tasks are made. The fourth step requires users to insert an Amharic query text using the button labeled የመረፊያ ቃል(፬) (search query). 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 የተገኘበትን ሰአት/ጊዜ አሳይ((፭) (show the time), 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 the first step. The sample result is shown in the text area of Figure 6.

Table 6 depicts transcription by hand, system transcription, and words that are correctly recognized by the system. Manual and system transcriptions are colored in blue to show the difference. The column ‘Transcription should be’ (Table 6), 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 6), 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 shows the interval upon which the query text በክልሎች is located. The square bracket ‘0:00:00.668000’ indicates that the search query term begins at 0 h, 0 min, and 0 s, whereas ‘0:00:10.070000’ indicates that the search query term ended at 0 h, 0 min, and 10 s.

Table 6. Search result description using news data.

Query Term	Transcription Should Be	Transcription by the System	Correctly Transcribed and Located
በክልሎች	segment00.wav [የኢንተርኔት አገልግሎትንም በተመለከተ በክልሎች በዞኖችና በአዲስአበባ 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, የአገልግሎት/አገልግሎት, ለማቋቋም

We can say that the word በክልሎች is located between 0 s and 10 s of the selected audio file. The last column of Table 6 shows a list of words በተመለከተ, በክልሎች, ለማቋቋም that 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 7, we could not find it at segment 00 (between 0 and 10 s) because it is not correctly recognized by the system. This requires further improvement of the ASR. Therefore, if the performance of the speech recognition is increased, decoding and system retrieval for every query term in a given speech could be located with its respective time frame more precisely.

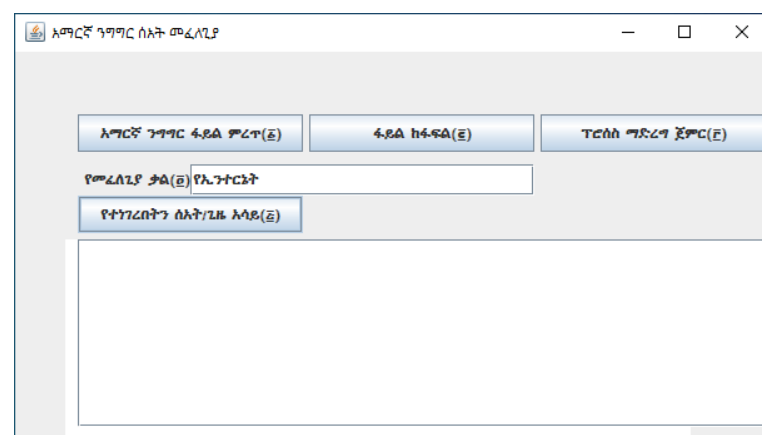


Figure 7. GUI showing unrecognized word using news query word.

6.8.2. Speech Search Using Bible Data

This section shows text-based STD, which was developed using ASR with a WER of 46%, tested with a spiritual domain (Bible speech) Amharic audio file. The result was obtained by interpolating the LMs of LVCSR and Bible LMs. The screenshot shown in Figure 8 depicts Amharic text-based STD GUI, which can accept Amharic query text (from the Bible domain) and displays where the speech is located in hours, minutes, and seconds. A detailed explanation of the retrieved result shown in Figure 8, obtained by using the query term **ፍሬ**, is shown in Table 7.

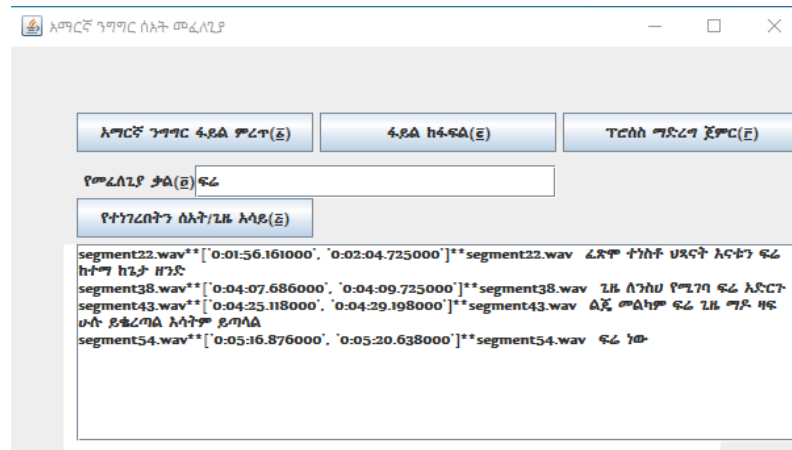


Figure 8. GUI for searching the speech using Bible query word.

Table 7. Search result description using the Bible data.

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 ልጄን ከግብጽ ጠራሁት የተባለው እንዲፈጸም ወደ ግብጽ ሄደ]	segment22.wav** ['0:01:56.161000', '0:02:04.725000']**segment22.wav [ፈጽሞ ተነስቶ ህጻኑንና እናቱን ፍሬ ከተማ ከጌታ ዘንድ]	ተነስቶ, እናቱን, ከጌታ, ዘንድ, ህጻኑንና/ህጻናት
ፍሬ	segment38.wav [እንግዲህ ለንስህ የሚገባ ፍሬ አድርጉ]	segment38.wav** ['0:04:07.686000', '0:04:09.725000']**segment38.wav [ጊዜ ለንስህ የሚገባ ፍሬ አድርጉ]	ለንስህ, የሚገባ, ፍሬ, አድርጉ
ፍሬ	segment43.wav [እንግዲህ መልካም ፍሬ የማያደርግ ዛፍ ሁሉ ይቈረጣል ወደ እሳትም ይጣላል]	segment43.wav** ['0:04:25.118000', '0:04:29.198000']**segment43.wav ልጄ መልካም ፍሬ ጊዜ ማዶ ዛፍ, ሁሉ ይቈረጣል እሳትም ይጣላል	መልካም, ፍሬ, ዛፍ, ሁሉ, ይቈረጣል, እሳትም, ይጣላል
ፍሬ	segment54.wav [እነሆም ሰማያት ተከፈቱ የእግዚአብሔርም መንፈስ]	segment54.wav** ['0:05:16.876000', '0:05:20.638000']**segment54.wav ፍሬ ነው	No word was recognized

From Table 7, for 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 when 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 ባሕርንግ was decoded as ባሕርን, which could be solved by finding the root word using a steamer.

6.8.3. Accuracy

In order to measure the accuracy, we selected a total of the 26 most and less frequently used words within a training set of LVCSR [41]. Most frequent words are words whose frequency are greater than or equal to 15 or words that exist more than 15 times, and the less frequent words are those that exist less than 15 times in the training set of the speech corpus. The *ATWV* of every selected query term were calculated, which showed an average *ATWV* of 85%. In our experimental evaluation, we understood that when the $N_{correct}$ is less and N_{true} is higher, the average *ATWV* will decrease and in reverse when the $N_{correct}$ and N_{true} values difference approach zero, the average *ATWV* will increase, which implies that the performance of the system will increase.

6.8.4. Efficiency

We measured the segmentation and transcription time of ASSTWQ. Table 8 shows the time it took to segment and transcribe unsegmented Amharic audio files from the broadcast and spiritual domains. Even if the time it takes to decode an unsegmented file is relatively similar to the length of the file, it is a one-time process. Once the segmentation and decoding are done, it takes a few seconds to search for a particular spoken word.

Table 8. Efficiency of ASSTWQ.

No.	Domains	Duration	Segmentation Time	Decoding Time
1	Broadcast (Amharic news speech file)	00:07:59	00:00:49	00:12:30
2	Spiritual (Amharic Bible speech file)	00:08:31	00:01:09	00:13:24

7. Conclusions

Searching for the location of a particular word with its respective time interval is a challenging task, particularly for a language such as Amharic, which has distinguishing characteristics such as glottal, palatal, and labialized consonants. This study aimed to investigate the development of a system that locates the time interval of a text word within an Amharic speech. To select the optimal segmentation to be applied for the development of text-based STD, ASR systems were developed using corpus from the spiritual domain. The performance of the systems was evaluated by test speech that is segmented using different segmentation methods: manual, phrase/word-like automatic, and sentence-like automatic. We found that the performance of the systems on sentence-like segmented test speech was much more similar to that of the manually segmented test speech. Therefore, we have proposed Amharic speech search using text word query (ASSTWQ) based on automatic sentence-like segmented test speech and using a previously developed speech corpus, which is in the broadcast news domain, together with the in-domain using the Bible domain speech corpus. The result showed that using the same domain for both the test and training speech corpus for LVCSR development performs better than using different domains. However, we also went further to check the performance of LVCSR by combining the training text corpus from the broadcast domain and the Bible text by interpolating the LMs and using the Bible as a test speech. We found that combining the LMs of two different domains showed better results than training LVCSR from one domain and testing with a speech from another domain. Finally, the developed ASSTWQ was tested with different query words that exactly locate the time interval in which the query text was located. In the future, this work can be extended in three ways. First, the technique of comparing the automatically and manually segmented speech can be applied to the training speech corpus by using the same methodology and technique. Second, searching on the text-based

STD can be done using phrase- and sentence-like context searching, which can handle homophones. Finally, since the Amharic language is a morphologically rich language, we extend the STD search using steamers both on the query word and the transcribed text to improve the effectiveness of the text-based STD.

Author Contributions: Conceptualization, S.T.A. and G.M.B.; methodology, S.T.A., T.A.A. and A.M.S.; software, S.T.A. and G.M.B.; validation, S.T.A.; writing—original draft preparation, G.M.B. and T.A.A.; writing—review and editing, T.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Postdoctoral Foundation of Zhejiang Normal University under Grant ZC304021941.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Categories of Amharic Consonants.

Manner of Articulation	Voicing	Labials	Alveolar	Palatals	Velars	LabioVelar	Glottals				
Stops	Voiceless	p	ፕ	T	ፐ	k	ክ	kwa	ኳ	Ax	ሶ
	Voiced	b	ብ	D	ድ	g	ግ	gwa	ግ		
	Glottalized	px	ፕጽ	Tx	ፐጽ	q	ቅ	qwa	ቅ		
Fricat-ives	Voiceless	f	ፍ	S	ሰ	sx	ሸ			H	ሀ
	Voiced	f	ፍ	z	ዝ	zx	ሸጾ				
	Glottalize			xx	ጽ					Hwa	ሻ
Affricat-ives	Voiceless				c	ች					
	Voiced				j	ጅ					
	Glottalized				cx	ቸ					
Nasals	Voiced	m	ም	N	ን	nx	ኝ				
Liquids	Voiced			L	ሌ						
Liquids	Voiced			R	ሮ						
Glides		w	ው			y	ይ				

Appendix B

Table A2. Shows Sample Core Characters Used in Amharic Writing System with their Seven Orders.

	Order						
	1st	2nd	3rd	4th	5th	6th	7th
	፩	ሀ	ሐ	ሰ	ሰ	ሰ	ሰ
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	ራ	ሮ	ሮ
s	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ

References

- Tejedor, J.; Toledano, D.T.; Ramirez, J.M.; Montalvo, A.R.; Alvarez-Trejos, J.I. The Multi-Domain International Search on Speech 2020 ALBAYZIN Evaluation: Overview, Systems, Results, Discussion and Post-Evaluation Analyses. *Appl. Sci.* **2021**, *11*, 8519. [\[CrossRef\]](#)
- Sakran, A.E.; Abdou, S.M.; Hamid, S.E.; Rashwan, M. A review: Automatic speech segmentation. *Int. J. Comput. Sci. Mob. Comput.* **2017**, *6*, 308–315.

3. Mary, L.; Deekshitha, G. *Searching Speech Databases: Features, Techniques and Evaluation Measures*; Springer: Cham, Switzerland, 2018.
4. Larson, M.; Jones, G.J. Spoken content retrieval: A survey of techniques and technologies. *Found. Trends Inf. Retr.* **2012**, *5*, 235–422. [[CrossRef](#)]
5. Kalantari, S. Improving Spoken Term Detection Using Complementary Information. Ph.D. Thesis, Queensland University of Technology, Brisbane, QLD, Australia, 2015.
6. Gündođdu, B. Keyword Search for Low Resource Languages. Ph.D. Thesis, Bogaziçi University, Istanbul, Turkey, 2017.
7. Tejedor, J.; Toledano, D.T.; Lopez-Otero, P.; Docio-Fernandez, L.; Montalvo, A.R.; Ramirez, J.M.; Peñagarikano, M.; Rodriguez-Fuentes, L.J. ALBAYZIN 2018 spoken term detection evaluation: A multi-domain international evaluation in Spanish. *EURASIP J. Audio Speech Music Process.* **2019**, *2019*, 1–37. [[CrossRef](#)]
8. Wang, X.; Zhang, P.; Na, X.; Pan, J.; Yan, Y. Handling OOVWords in Mandarin Spoken Term Detection with an Hierarchical n-Gram Language Model. *Chin. J. Electron.* **2017**, *26*, 1239–1244. [[CrossRef](#)]
9. Woldeyohannis, M.M.; Besacier, L.; Meshesha, M. Amharic speech recognition for speech translation. In Proceedings of the TALALF 2016: Traitement Automatique des Langues Africaines (Text and Speech) JEP-TALN-RECITAL 2016 Workshop, Paris, France, 4 July 2016; pp. 114–124.
10. Van Heerden, C.; Karakos, D.; Narasimhan, K.; Davel, M.; Schwartz, R. Constructing sub-word units for spoken term detection. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 5780–5784.
11. Mousa, A.E.D.; Kuo, H.K.J.; Mangu, L.; Soltan, H. Morpheme-based feature-rich language models using deep neural networks for LVCSR of Egyptian Arabic. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 8435–8439.
12. Malik, M.; Malik, M.K.; Mehmood, K.; Makhdoom, I. Automatic speech recognition: A survey. *Multimed. Tools Appl.* **2021**, *80*, 9411–9457. [[CrossRef](#)]
13. Gales, M.; Young, S. The application of hidden Markov models in speech recognition. *Found. Trends® Signal Process.* **2008**, *1*, 195–304. [[CrossRef](#)]
14. Tachbelie, M.Y. Morphology-Based Language Modeling for Amharic. Ph.D. Thesis, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky, Hamburg, Germany, 2010.
15. Che, D.; Shafer, T.; Tian, P. Classification of endangered languages using decision tree based algorithms. In Proceedings of the 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Guilin, China, 29–31 July 2017; pp. 1814–1821.
16. Gereme, F.; Zhu, W.; Ayall, T.; Alemu, D. Combating fake news in “low-resource” languages: Amharic fake news detection accompanied by resource crafting. *Information* **2021**, *12*, 20. [[CrossRef](#)]
17. Abate, S.T.; Menzel, W. Syllable-based speech recognition for Amharic. In Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, Prague, Czech Republic, 28–29 June 2007; pp. 33–40.
18. Biadgigne, Y.; Smaïli, K. Parallel Corpora Preparation for English-Amharic Machine Translation. In Proceedings of the International Work-Conference on Artificial Neural Networks, Virtual, 16–18 June 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 443–455.
19. Abate, S.T. Automatic Speech Recognition for Amharic. Ph.D. Thesis, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky, Hamburg, Germany, 2006.
20. Pala, M.; Parayitam, L.; Appala, V. Real-time transcription, keyword spotting, archival and retrieval for telugu TV news using ASR. *Int. J. Speech Technol.* **2019**, *22*, 433–439. [[CrossRef](#)]
21. Chaudhary, A.; Akshatha, K.; Kodlekere, K.; Prasad, S.J. Keyword based indexing of a multimedia file. In Proceedings of the 2017 IEEE International Symposium on Multimedia (ISM), Taichung, Taiwan, 11–13 December 2017; pp. 573–576.
22. Tamiru, R.M.; Abate, S.T. Sentence-Level Automatic Speech Segmentation for Amharic. In Proceedings of the Sixth International Congress on Information and Communication Technology, London, UK, 25–26 February 2021; Springer: Berlin/Heidelberg, Germany, 2022; pp. 477–485.
23. Tukeyev, U.; Karibayeva, A.; Zhumanov, Z.h. Morphological segmentation method for Turkic language neural machine translation. *Cogent Eng.* **2020**, *7*, 1856500. [[CrossRef](#)]
24. Abate, M.; Assabie, Y. Development of Amharic morphological analyzer using memory-based learning. In Proceedings of the International Conference on Natural Language Processing, Warsaw, Poland, 17–19 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 1–13.
25. Meinedo, H.; Neto, J.P. Automatic speech annotation and transcription in a broadcast news task. In Proceedings of the ISCA Workshop on Multilingual Spoken Document Retrieval, Hong Kong, China, 4–5 April 2003; pp. 95–100.
26. Kim, J.Y.; Liu, C.; Calvo, R.A.; McCabe, K.; Taylor, S.C.; Schuller, B.W.; Wu, K. A comparison of online automatic speech recognition systems and the nonverbal responses to unintelligible speech. *arXiv* **2019**, arXiv:1904.12403.
27. Sharma, P.; Rajpoot, A.K. Automatic identification of silence, unvoiced and voiced chunks in speech. *J. Comput. Sci. Inf. Technol. CS IT* **2013**, *3*, 87–96.
28. Silber-Varod, V.; Geri, N. Can automatic speech recognition be satisfying for audio/video search? Keyword-focused analysis of Hebrew automatic and manual transcription. *Online J. Appl. Knowl. Manag.* **2014**, *2*, 104–121.

29. Hassen, A.; Atnafu, S. Develop an Audio Search Engine for Amharic Speech Web Resources. Master's Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2017.
30. Al Laham, M.N.; Ayass, I.; Ghareeb, M.; El-Bazzal, Z.; Raad, M. Audio indexing for YouTube. In Proceedings of the 2015 Fifth International Conference on Digital Information and Communication Technology and its Applications (DICTAP), Beirut, Lebanon, 29 April–1 May 2015; pp. 111–114.
31. Arisoy, E.; Can, D.; Parlak, S.; Sak, H.; Saraçlar, M. Turkish broadcast news transcription and retrieval. *IEEE Trans. Audio Speech Lang. Process.* **2009**, *17*, 874–883. [[CrossRef](#)]
32. WordProject. Bibles. Available online: <https://www.wordproject.org/bibles/index.htm> (accessed on 1 September 2021).
33. Bible. Available online: <https://www.bible.com/bible/1260/MAT.10.NASV> (accessed on 10 August 2021).
34. Anberbir, T.; Gasser, M.; Takara, T.; Yoon, K.D. Grapheme-to-phoneme conversion for Amharic text-to-speech system. In Proceedings of the Conference on Human Language Technology for Development, Bibliotheca Alexandrina, Alexandria, Egypt, 2–5 May 2011; pp. 68–73.
35. Abate, S.T.; Menzel, W.; Tafila, B. An amharic speech corpus for large vocabulary continuous speech recognition. In Proceedings of the Ninth European Conference on Speech Communication and Technology, Lisbon, Portugal, 4–8 September 2005; pp. 1601–1604.
36. Abate, S.T.; Melese, M.; Tachbelie, M.Y.; Meshesha, M.; Atnafu, S.; Mulugeta, W.; Assabie, Y.; Abera, H.; Ephrem, B.; Abebe, T.; et al. Parallel Corpora for bi-lingual English-Ethiopian Languages Statistical Machine Translation. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; Association for Computational Linguistics: Santa Fe, NM, USA, 2018; pp. 3102–3111.
37. CMUSphinx. Available online: <https://cmusphinx.github.io/> (accessed on 1 September 2021).
38. Ahmed, S.; Chowdhury, A.R.; Fawaz, K.; Ramanathan, P. Preech: A System for Privacy-Preserving Speech Transcription. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 20), Virtual, 12–14 August 2020; pp. 2703–2720.
39. Gaikwad, S.K.; Gawali, B.W.; Yannawar, P. A review on speech recognition technique. *Int. J. Comput. Appl.* **2010**, *10*, 16–24. [[CrossRef](#)]
40. Miller, D.R.; Kleber, M.; Kao, C.L.; Kimball, O.; Colthurst, T.; Lowe, S.A.; Schwartz, R.M.; Gish, H. Rapid and accurate spoken term detection. In Proceedings of the Eighth Annual Conference of the International Speech Communication Association, Antwerp, Belgium, 27–31 August 2007; pp. 314–317.
41. Gosztolya, G.; Tóth, L. Spoken term detection based on the most probable phoneme sequence. In Proceedings of the 2011 IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMII), Smolenice, Slovakia, 27–29 January 2011; pp. 101–106.